

For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex LIBRIS
UNIVERSITATIS
ALBERTAENSIS





Digitized by the Internet Archive
in 2020 with funding from
University of Alberta Libraries

<https://archive.org/details/Nabona1971>

THE UNIVERSITY OF ALBERTA

A DIRECT LOAD-FLOW METHOD



by

Narcís Nabona

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

SPRING, 1971

UNIVERSITY OF ALBERIA
FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled A Direct Load-Flow Method submitted by Narcís Nabona in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

This thesis presents a new approach to the digital solution of the power system load-flow problem.

The most popular approach in use today is the Newton-Raphson method which is fast but uses a considerable amount of core memory.

The method proposed here is a modification of the Newton-Raphson method, which, although slower in convergence, can save considerable time in solution as it eliminates the need for recalculation of the coefficients of the linearized system at each iteration.

The core requirements are only slightly higher than Newton-Raphson's.

ACKNOWLEDGMENTS

The author is appreciative of the helpful advice given to him by his supervisor, Prof. D. H. Kelly, who made this thesis possible.

He would also like to thank all his friends, who have made the English of this thesis more presentable.

TABLE OF CONTENTS

	<u>page</u>
TABLE OF CONTENTS	I
LIST OF FIGURES	IV
LIST OF SYMBOLS	V
I THE LOAD-FLOW PROBLEM	
1.1 Definition	1
1.2 Methods of solution	3
1.2.1 Classification of the methods of solution	8
1.2.2 Features of the most important methods	10
1.3 Brief scope of the thesis	11
II THE NEWTON-RAPHSON METHOD	
2.1 Description	13
2.2 Numerical properties	17
2.3 Analysis of used time	19
III THE PROPOSED METHOD	
3.1 An iterative method for solving a set of nonlinear quadratic equations	22
3.1.1 The new iterative method in the one-dimensional case	24
3.1.2 The Newton-Raphson method in the one-dimensional case	30
3.1.3 Comparison of the convergence of the new method with an one-dimensional quadratic system, with that of Newton-Raphson's	31
3.2 Developement of the method	32
3.3 The bases for convergence	37
3.4 Comparison between the Newton-Raphson method and the proposed method	39
3.4.1 Difference between the concept of increment of voltage in the Newton-Raphson method and in the proposed method	41
3.4.2 Calculation of the elements of the Jacobian	43
3.5 Outstanding features of the proposed method	44
IV THE PROPOSED METHOD USING TRIANGULATION WITH PIVOTING FOR SOLVING THE LINEARIZED SYSTEM AT EACH ITERATION. RESULTS AND CRITICISM	
4.1 Numerical results of the proposed method	45
4.1.1 Explanations on the computer program used	45
4.1.2 Test system used	46
4.1.3 Numerical results	46
4.2 The effect of acceleration factors	48
4.3 Combination of the Newton-Raphson method and the proposed method using triangulation with pivoting	49
4.4 The weak points in the proposed method	50

4.5	The weak points in the Newton-Raphson method	52
4.6	Possible improvements	52
4.6.1	The efficiency of the steps of a method with respect to time	53
4.6.2	Advantage of the proposed method over the Newton-Raphson method	53
V	THE THEORETICAL BASIS FOR THE USE OF THE GAUSS-SEIDEL ITERATIVE TECHNIQUE FOR SOLVING THE LINEARIZED SYSTEM IN THE PROPOSED METHOD	
5.1	Feasibility of the solution	55
5.1.1	Requirements for convergence	55
5.1.2	Characteristics of the matrix of coefficients of the linearized system	56
5.1.2.1	The elimination of the diagonal of L^{III}	59
5.1.2.2	Characteristics of the method	62
5.2	The application of an iterative technique for solving the linearized system	63
5.2.1	Difference between iteration and run	63
5.2.2	Tendency of the results obtained in the runs	65
VI	RESULTS OF THE PROPOSED METHOD USING THE GAUSS-SEIDEL ITERATIVE TECHNIQUE FOR SOLVING THE LINEARIZED SYSTEM	
6.1	Analysis of the results of an exact, and an approximate solution of the linearized system in the proposed method	67
6.1.1	Results for the 14 bus test system	68
6.1.2	Results for the 30 bus test system	73
6.1.3	Results for the 57 bus test system	75
6.2	Comparison between the results in the three systems and conclusions	77
VII	IMPROVEMENTS IN THE PROPOSED METHOD	
7.1	The use of acceleration factors in the solution of the linearized system by the Gauss-Seidel technique	81
7.1.1	Acceleration formula used	
7.2	Analysis of the effect of acceleration factors in the solution of the linearized system	82
7.3	Comments on the use of acceleration factors	87
7.4	Iterative approximate solution and switch to an exact solution and to the Newton-Raphson method	89
VIII	CONCLUSION	
8.1	Inconveniences and advantages of the proposed method	95
8.2	Real load-flow programs and research load-flow programs	97
8.3	Suggestions for further research on the proposed method	99
	REFERENCES	101
	APPENDIX 1 . AN ITERATIVE METHOD FOR THE SOLUTION OF A SET OF QUADRATIC NONLINEAR EQUATIONS	
	APPENDIX 2 . THE CALCULATION OF THE Y_{bus} MATRIX	

APPENDIX 3 . THE SUBROUTINE FOR THE IMPLEMENTATION OF THE PROPOSED
METHOD AND THE NEWTON-RAPHSON METHOD

APPENDIX 4 . OTHER SUBROUTINES USED

APPENDIX 5 . THE SOLUTION OF THE LOAD-FLOW PROBLEMS

APPENDIX 6 . OPERATION WITH MORE THAN ONE SLACK BUS

LIST OF FIGURES

		<u>page</u>
Figure 1.1	SIMPLE EXAMPLE	6
Figure 2.1	PERCENTAGE OF TIME FOR STAGES OF COMPUTATION VS. NUMBER OF BUSES	20
Figure 2.2	TIME PER ITERATION VS. NUMBER OF BUSES	20
Figure 7.1	ERROR IN THE REACTIVE COMPONENT OF THE INCREMENT IN VOLTAGE, DFF, VS. TIME	83
Figure 7.2	ERROR IN THE REACTIVE COMPONENT OF THE INCREMENT IN VOLTAGE, DFF, VS. TIME	92

TABLE OF SYMBOLS

B	susceptance of the bus admittance matrix, ($Y_{pq} = G_{pq} + jB_{pq}$)
CIR, CII	real and reactive vectors of currents at the buses
DCR, DCI	vectors of the real and reactive differences between the true currents injected at the buses, and those calculated with the initial estimation of the voltages
DFT	vector of increments of voltage with respect to the initial ones
E, F	vectors of real and reactive bus voltages
e, f	components of the vectors E, F
EPS	tolerance or precision
FRL, FIM	acceleration factors for the real and imaginary components of the voltage changes
G	reactance of the bus admittance matrix, ($Y_{pq} = G_{pq} + jB_{pq}$)
$g(x_i)$	iterative one-dimensional function, ($x_{i+1} = g(x_i)$)
I	vector of currents
ISW	number of the iteration in which there is a switch to an exact method of solution of the linearized system in the proposed load-flow method
ITE	number of iteration
J, j_{ik}	Jacobian and one of its elements
L, l_{ik}	matrix of coefficients of the linearized system of equations in the proposed method, and one of its elements
P, Q	vectors of real and reactive injected powers at the buses
$q(x)$	one-dimensional quadratic function
SMX	maximum mismatch in power with respect to the scheduled powers
TMX	maximum change in voltage between two successive iterations
Y_{bus}	bus admittance matrix
Z_{bus}	bus impedance matrix
λ	Lipschitz constant, ($0 \leq \lambda < 1$)

I. THE LOAD-FLOW PROBLEM

1.1 DEFINITION

The load flow problem is the problem of determining the voltages and power flows in a power system for a given set of loads and system connections. It is necessary to know in advance, the voltages and generator outputs required to carry the loads in a satisfactory manner.

The terminal or bus conditions in a network of distribution are likely to change continuously. Indeed, the demand of power by the users is variable, and the generation has to be changed accordingly. Different patterns of load at the load buses, require different patterns of generation at the generation buses. However, not all patterns of generation are possible because of limitations in bus voltages and line current carrying capacity.

Each pattern of generation, for a given pattern of loads, implies a cost of generation plus a certain amount of power losses in the network. If an optimum economical operation is to be implemented, several patterns of generation have to be tried, so that the best pattern can be found.

Similarly, in order to ensure that the powers carried and the voltages are within the safety levels, a number of studies have to be made with different generation patterns. This also must be done for testing the electric stability of the power system when operated with a specified pattern of loads and generations.

Since the power demand is increasing, new power stations and

new lines of interconnection have to be built up. Preliminary planning studies have to be carried out to check the feasibility of the new supply as far as stability, safety levels and economic optimality are concerned. The performance of a power system is known completely if we know its load-flow solution, and from this stems the importance of the load-flow studies.

The essential problem is that of determining the node voltages in the system as, once these are known, the power flows may be calculated by use of equation (1.1),

$$P_{pq} - jQ_{pq} = E_p^*(E_p - E_q)y_{pq} + E_p^*E_p \frac{y'_{pq}}{2} \quad (1.1)$$

where p and q stand for the code number given to two of the buses of the network. The current flowing into the system at bus "p", can be calculated by equation (1.2)

$$I_p = \frac{P_p - jQ_p}{E_p^*} \quad (1.2)$$

A single phase representation is enough because all problems dealt with are symmetric and balanced.

Several representations of the network are possible. Each representation has several associated methods of solution of the load flow problem.

In 1929 the alternating current network analyzer, which is a special purpose analogue computer, was created. This device was used, among other studies, for the solution of load-flow problems.

The network analyzer severely limits the size of problem which may be handled, requiring a simplified representation of the system, in many cases. However until the mid 1950s digital computers were not satisfactory due to core limitations. From the late 1950s onward, the progress in digital computer performances has kept pace with the increasing complexity and size of power system problems. Nowadays digital computers have replaced alternating current network analyzers as the tool for the power system engineer. At the same time the amount of memory required and the computer time spent have become the indications of goodness of a load flow method. This is because the load-flow problem, as it has been shown before, has to be solved many times, and both the amount of computer memory and time used affect costs.

1.2 METHODS OF SOLUTION

Different mathematical descriptions of the network result in different load-flow equations. The first method used for digital solution implemented the loop frame of reference in admittance form. However, this form takes a longer data preparation than the ones that use the bus frame of reference, in either admittance or impedance form.

Currents or voltages can be chosen as independent variables. If currents are selected, the impedance form must be used. Equation (1.2) along with (1.3) are used

$$E_p = \sum_q Z_{bus\ pq} I_q + E_p^R \quad (1.3)$$

where E_p^R is the voltage at the reference bus.

If the admittance form is selected, the equations used have the voltages as independent variables, and several equations such as (1.4) and (1.5) could be used depending on the method of solution applied

$$E_p = \frac{1}{Y_{pp}} \left(\frac{P_p - jQ_p}{E_p^*} - \sum_{q \neq p} Y_{pq} E_q \right) \quad p = 1, \dots, N, \quad p \neq s \text{ ("s" means slack bus, see below)} \quad (1.4)$$

$$P_p = \sum_q \{ e_p (e_p G_{pq} + f_q B_{pq}) + f_p (f_q G_{pq} - e_q B_{pq}) \}$$

$$Q_p = \sum_q \{ f_p (e_q G_{pq} + f_q B_{pq}) - e_p (f_q G_{pq} - e_q B_{pq}) \} \quad (1.5)$$

where ($Y_{pq} = G_{pq} - jB_{pq}$)

There are four quantities associated with each bus. These are the real and reactive power, and the real and reactive voltage. Only two of the quantities are known at each bus before the solution of the problem. Depending on which quantities are known, the bus is called differently and processed in the appropriate way. We can have the following kinds of buses.

Power buses

- generation buses
- load buses
- generation plus load buses

Voltage buses

- voltage controlled buses
- slack buses

Power buses are the ones at which the powers flowing into the system or out of the system are known, and the active and

and reactive voltage are unknown. If the power at the bus is flowing into the system, we call it a generation bus. If the power is flowing out of the system we call it a load bus. When there is generation and load at the same bus, it is called generation plus load bus; in general the generation and the load in such buses are not balanced. Therefore at such buses a certain amount of power is flowing into the system or out of it. The powers have a positive sign when flowing in, and negative when flowing out.

Voltage buses are the ones where either the voltage magnitude is constant or both components, the real and the reactive voltage are constant. The buses at which the voltage magnitude is constant can change their active and reactive voltages provided the voltage magnitude is kept constant. In such buses the real power is known and constant, and the reactive voltage varies so that the voltage magnitude is constant. These are called voltage controlled buses. When both components of the voltage are known, the bus is called a slack bus. In that case neither the real power nor the reactive power are known before the solution of the load-flow problem.

Trial and error techniques can be used to account for such special cases as limited available reactive power at voltage controlled buses, tap-changing under load transformers and specified net interchange of power. Although it takes a longer time, the solution is always reached.

The losses in a power system are a function of the node or bus voltages, which are unknown before the load-flow solution.

Since the generations have to match the loads plus the losses, there must be at least one slack bus, with an indeterminate power inflow, in order to generate, at that node, as much power as needed to balance the losses.

A very simple example is given now to show the different types of buses and the equations to be solved. The admittance form and equation (5) are chosen for the mathematical formulation. The problem consists of four buses as shown in figure 1-1.

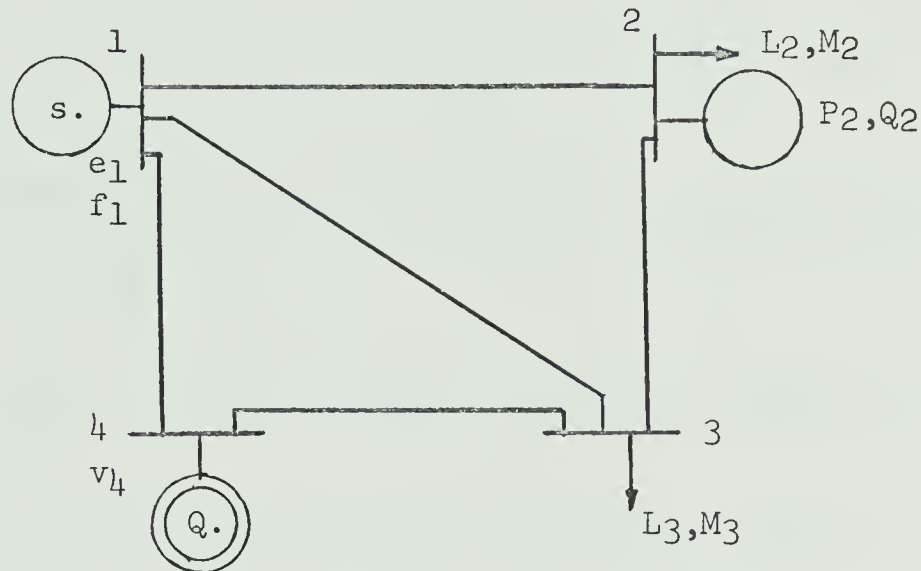


Figure 1.1 SIMPLE EXAMPLE

Bus 1 is a slack bus. Its voltages $\bar{e}_1$, $\bar{f}_1$ are fixed and known. Its powers P_1 , Q_1 , are unknown and will be computed at the end of the process, once the voltages at all other nodes are known, by means of equation (1.5).

Bus 2 is a generation plus load bus. Its voltages e_2 , f_2 are unknown and are found by solution of the load-flow problem. The generation at this node is $P_2 - jQ_2$ and its load $L_2 - jM_2$. The

power flow into the system is thus: $(P_2 - L_2) - j(Q_2 - M_2)$. The way in which equation (1.5) is applied in this case is the following,

$$\begin{aligned} \bar{P}_2 - \bar{L}_2 = & e_2(\bar{e}_1\bar{G}_{21} + \bar{f}_1\bar{B}_{21}) + f_2(\bar{f}_1\bar{G}_{21} - \bar{e}_1\bar{B}_{21}) + \\ & + e_2(e_3\bar{G}_{23} + f_3\bar{B}_{23}) + f_2(f_3\bar{G}_{23} - e_3\bar{B}_{23}) \end{aligned} \quad (1.6)$$

(the symbols with a dash are known quantities)

$$\begin{aligned} \bar{Q}_2 - \bar{M}_2 = & f_2(\bar{e}_1\bar{G}_{21} + \bar{f}_1\bar{B}_{21}) - e_2(\bar{f}_1\bar{G}_{21} - \bar{e}_1\bar{B}_{21}) + \\ & + f_2(e_3\bar{G}_{23} + f_3\bar{B}_{23}) - e_2(f_3\bar{G}_{23} - e_3\bar{B}_{23}) \end{aligned} \quad (1.7)$$

Bus 3 is a load node. The power outflow or negative in-flow is $-(L_3 - jM_3)$

$$\begin{aligned} -\bar{L}_3 = & e_3(\bar{e}_1\bar{G}_{31} + \bar{f}_1\bar{B}_{31}) + f_3(\bar{f}_1\bar{G}_{31} - \bar{e}_1\bar{B}_{31}) + \\ & + e_3(e_2\bar{G}_{32} + f_2\bar{B}_{32}) + f_3(f_2\bar{G}_{32} - e_2\bar{B}_{32}) + \\ & + e_3(e_4\bar{G}_{34} + f_4\bar{B}_{34}) + f_3(f_4\bar{G}_{34} - e_4\bar{B}_{34}) \end{aligned} \quad (1.8)$$

$$\begin{aligned} -\bar{M}_3 = & f_3(\bar{e}_1\bar{G}_{31} + \bar{f}_1\bar{B}_{31}) - e_3(\bar{f}_1\bar{G}_{31} - \bar{e}_1\bar{B}_{31}) + \\ & + f_3(e_2\bar{G}_{32} + f_2\bar{B}_{32}) - e_3(f_2\bar{G}_{32} - e_2\bar{B}_{32}) + \\ & + f_3(e_4\bar{G}_{34} + f_4\bar{B}_{34}) - e_3(f_4\bar{G}_{34} - e_4\bar{B}_{34}) \end{aligned}$$

Bus 4 is a voltage controlled bus. The voltage magnitude is kept at the value v_4 , hence,

$$v_4^2 = e_4^2 + f_4^2 \quad (1.9)$$

the reactive power needed will be calculated once the solution has been reached by means of equation (1.5). As for the active power, we do not have in this case any power inflow at that node, since there is no real power generation nor real load, therefore,

$$\begin{aligned} 0 = & e_4(\bar{e}_1\bar{G}_{41} + \bar{f}_1\bar{B}_{41}) + f_4(\bar{f}_1\bar{G}_{41} - \bar{e}_1\bar{B}_{41}) + \\ & + e_4(e_3\bar{G}_{43} + f_3\bar{B}_{43}) + f_4(f_3\bar{G}_{43} - e_3\bar{B}_{43}) \end{aligned} \quad (1.10)$$

These six equations have to be solved to find $e_2, f_2, e_3, f_3, e_4, f_4$. As these equations are nonlinear equations, an iterative technique must be used. Several methods could be used for solving this set of nonlinear equations by means of a digital computer.

1.2.1 CLASSIFICATION OF THE METHODS OF SOLUTION

As a classification of the methods of solution of the load-flow problem, that given by Laughton and Humphrey Davies (1) of direct load flow methods and iterative load-flow methods, could be taken.

Sasson and Jaimes (2), came up later with a new classification, (nodal admittance matrix methods, variational matrix methods, and nodal impedance matrix methods), objecting to the ambiguity of the former classification as the direct methods are iterative in nature.

This objection could be refuted by changing the name "iterative methods". Since all of the so called iterative methods correct

the voltage of only one bus at a time, they could be called "bus by bus" methods, and this name will be used in the rest of this section.

In a recent compendium of computer methods in power system analysis, Stagg and El-Abiad (3), describe the most usual load-flow methods without any classification. However, the methods presented can all be fitted into the classification, of direct methods - bus by bus methods.

bus by bus methods	Gauss iterative method using Y_{bus}
	Gauss iterative method using Z_{bus}
	Gauss-Seidel method using Y_{bus}
	Gauss-Seidel method using Z_{bus}
	Relaxation method using Y_{bus}
direct methods	Newton-Raphson method using Y_{bus}
	Gauss iterative method using Y_{loop}

A unified nonlinear programming approach to solve the load-flow, minimum loss, and economic dispatch problems has been shown to be possible by Albert M. Sasson (4).

All methods start with a first assumption of the bus voltages. Bus by bus methods use an algorithm which then computes a better approximation for each bus on a single bus basis. The order of computation follows the given numbers of the buses in Gauss-Seidel methods and Gauss iterative methods. In the relaxation method, the only bus whose voltage is corrected at each step, is that with

the largest residual, which is a measure of the mismatch in power.

In the direct methods the correction of the voltages come from an operation which uses all the available information. This operation is the solution of a linear system of equations in the Newton-Raphson method, and is the calculation of several vectors and matrix multiplications in the case of the Gauss iterative method using Y_{loop} .

1.2.2 FEATURES OF THE MOST IMPORTANT METHODS

There is not a best method in load-flow because there can always exist a specified power system for which one of the methods, which does not work as well as the other methods with other problems, gives better results than the others. Moreover, even by changing the initial guess in the same load-flow problem, it is possible to find different performances of the methods. All methods have advantages and disadvantages and, in most of the cases, the selection of one method or another depends on the weight placed on every advantage and disadvantage.

Bus by bus methods in general have the disadvantage that in order to have a small mismatch in power, they have to iterate until a considerably smaller increment in voltage is reached.

The early method of Ward and Hale (5), has been improved by other more sophisticated bus by bus methods. In their book (3), Stagg and El-Abiad, (p. 328), show how the Gauss-Seidel method using Y_{bus} takes the lowest time per iteration. But as can be seen later, (pp. 330 and 331), this advantage is balanced by the high

number of iterations for solution.

As far as time is concerned, it can be seen that Newton-Raphson method yields the best results, but there are other factors to account for. One of these is the computing time for both processing system data in order to obtain the parameters for the iterative solution, and modifying network data and introducing operating changes. In this regard the methods which use the bus admittance form are better off than any other form, because, provided that mutual coupling is not considered, the computation and changes of Y_{bus} , are very easy and straightforward.

There is still another factor, the storage required. The Gauss-Seidel method using Y_{bus} needs less computer memory than Newton-Raphson method using Y_{bus} . Whether the extra storage is offset by the savings of time by the Newton-Raphson method, has to be decided by the utility companies which use such methods.

Tinney and Hart, (6), have obtained some improvements in the practical application of the Newton-Raphson method to real power systems. Their work along with other studies, such as the work by Feingold and Spohn by Electricite de France, (7), seems to indicate that Newton-Raphson is the method most commonly employed by the big utility companies.

1.3 BRIEF SCOPE OF THE THESIS

An improvement in time with respect to the Newton-Raphson method has been found.

A different approach to the solution of the set of nonlinear equations (1.5), leads to a different iterative method of solution.

The main features of this method are:

- a) The calculation, once only, of the non-diagonal coefficients of the linearized set of equations. With the Newton-Raphson method this calculation must be made for each iteration.
- b) The feasibility of the implementation of fast approximate solution of the linearized system.

One of the advantages over the Newton-Raphson method is that not much time is spent during the early stages for the solution of the linearized system, since it is going to yield results, which are still far from the true solution. An exact time consuming method is not implemented until the voltages are close to the final solution. This idea had been already suggested by Van Ness in 1959, (8), but it had not been developed.

II. THE NEWTON-RAPHSON METHOD

2.1 DESCRIPTION

As the Newton-Raphson method is presently the most popular method in use, it will be described in some detail. Alternative methods may then be compared, with regard to speed and accuracy, with the Newton-Raphson method.

The Newton-Raphson iterative technique for finding the root of a function is used by the Newton-Raphson method for solving the set of equations (1.5). This method, as it is known to the power systems engineers, is nothing but the generalization of the Newton-Raphson technique for finding roots of a one dimensional function, to a multidimensional space.

The set of equations (1.5), may be written in a more general fashion (2.1)

$$\begin{aligned} P_p &= g_p(e_1, e_2, \dots, e_N, f_1, f_2, \dots, f_N) \\ Q_p &= h_p(e_1, e_2, \dots, e_N, f_1, f_2, \dots, f_N) \end{aligned} \quad (2.1)$$
$$p = 1, \dots, N$$

If the set of functions g_p and h_p are expanded about the point $(e^0_1, e^0_2, \dots, e^0_N, f^0_1, f^0_2, \dots, f^0_N)$ in a $2 \times N$ dimensional coordinate system, in a Taylor series, and drop the second order derivatives and those of higher order, then

$$P_p = g_p(e^0_1, e^0_2, \dots, e^0_N, f^0_1, f^0_2, \dots, f^0_N) +$$

$$\begin{aligned}
 & + \left. \frac{\delta g_p}{\delta e_1} \right|_o (e_1 - e^o_1) + \dots + \left. \frac{\delta g_p}{\delta e_2} \right|_o (e_i - e^o_i) + \dots + \left. \frac{\delta g_p}{\delta e_N} \right|_o (e_N - e^o_N) + \\
 & + \left. \frac{\delta g_p}{\delta f_1} \right|_o (f_1 - f^o_1) + \dots + \left. \frac{\delta g_p}{\delta f_i} \right|_o (f_i - f^o_i) + \dots + \left. \frac{\delta g_p}{\delta f_N} \right|_o (f_N - f^o_N) + \varepsilon
 \end{aligned}$$

$$\begin{aligned}
 Q_p &= h_p(e^o_1, e^o_2, \dots, e^o_N, f^o_1, f^o_2, \dots, f^o_N) + \\
 & + \left. \frac{\delta h_p}{\delta e_1} \right|_o (e_1 - e^o_1) + \dots + \left. \frac{\delta h_p}{\delta e_i} \right|_o (e_i - e^o_i) + \dots + \left. \frac{\delta h_p}{\delta e_N} \right|_o (e_N - e^o_N) + \\
 & + \left. \frac{\delta h_p}{\delta f_1} \right|_o (f_1 - f^o_1) + \dots + \left. \frac{\delta h_p}{\delta f_i} \right|_o (f_i - f^o_i) + \dots + \left. \frac{\delta h_p}{\delta f_N} \right|_o (f_N - f^o_N) + \varepsilon
 \end{aligned}$$

$i \neq s$, (S slack buses, see App. 6)

Calling

$$P^o_p = g_p(e^o_1, e^o_2, \dots, e^o_N, f^o_1, f^o_2, \dots, f^o_N)$$

$$Q^o_p = h_p(e^o_1, e^o_2, \dots, e^o_N, f^o_1, f^o_2, \dots, f^o_N)$$

$$j^o_{pi}^{I} = \left. \frac{\delta g_p}{\delta e_i} \right|_o$$

$$j^o_{pi}^{II} = \left. \frac{\delta g_p}{\delta f_i} \right|_o$$

$$j^o_{pi}^{III} = \left. \frac{\delta h_p}{\delta e_i} \right|_o$$

$$j^o_{pi}^{IV} = \left. \frac{\delta h_p}{\delta f_i} \right|_o$$

where the subscript "o" means at the point $(e^o_1, \dots, e^o_N, f^o_1, \dots, f^o_N)$

Equation (2.2) can be rewritten in matrix form

$$\begin{bmatrix} \dots \\ P_p - P^o_p \\ \dots \\ \text{---} \\ \dots \\ Q_p - Q^o_p \\ \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots & | & \dots & \dots & \dots \\ \dots & j^o_{pi}^I & \dots & | & \dots & j^o_{pi}^I & \dots \\ \dots & \dots & \dots & | & \dots & \dots & \dots \\ \text{---} & \text{---} & \text{---} & | & \text{---} & \text{---} & \text{---} \\ \dots & \dots & \dots & | & \dots & \dots & \dots \\ \dots & j^o_{pi}^{III} & \dots & | & \dots & j^o_{pi}^{IV} & \dots \\ \dots & \dots & \dots & | & \dots & \dots & \dots \end{bmatrix} \times \begin{bmatrix} \dots \\ e_i - e^o_i \\ \dots \\ \text{---} \\ \dots \\ f_i - f^o_i \\ \dots \end{bmatrix}$$

$p = 1, \dots, N ; \quad p \neq s$
 $i = 1, \dots, N ; \quad i \neq s, \quad (S \text{ slack buses})$

(2.3)

It must be pointed out that equation (2.3) is an approximate equation because the second order derivatives and those of higher order, have been neglected. It can be noticed that no equations for the slack buses are taken, (see appendix 6 for more than one slack bus).

As a result of the first solution of equation (2.3) an increment of voltage is obtained, which will bring the voltages to their new value. Indicating the first increment by $\Delta^1 e_i$ and $\Delta^1 f_i$

$$\begin{aligned}
 \Delta^1 e_i &= e_i - e^o_i = e^1_i - e^o_i \\
 \Delta^1 f_i &= f_i - f^o_i = f^1_i - f^o_i
 \end{aligned}$$

(2.4)

The superscript "1" is used to indicate the voltages after the first solution of (2.3), or first iteration. Of course

$$\begin{aligned}
 e^1_i &= e^o_i + \Delta^1 e_i \\
 f^1_i &= f^o_i + \Delta^1 f_i
 \end{aligned}$$

(2.5)

It can be said that, in a general iteration, the superscript of the increment in voltage is the number of the iteration whereas the superscript of the increment in power is that of the former iteration. The power increments are calculated with the voltages obtained in the iteration before, and the same for the terms of the jacobian, $j_{pi}^k I$, $j_{pi}^k II$, etc., where "k" indicates that these terms are calculated with the voltages e_i^k , f_i^k , obtained at the "kth" iteration. The general form of equation (2.3), that which would yield the "k+1" set of voltages, at the "k+1" iteration, is the next

$$\begin{bmatrix} \Delta^{kP} \\ \text{---} \\ \Delta^{kQ} \end{bmatrix} = \begin{bmatrix} J^k I & | & J^k II \\ \text{---} & | & \text{---} \\ J^k III & | & J^k IV \end{bmatrix} \times \begin{bmatrix} \Delta^{k+1e} \\ \text{---} \\ \Delta^{k+1f} \end{bmatrix} \quad (2.6)$$

The same process is followed at each iteration. When a new set of voltages e_i^k, f_i^k are obtained, the functions g_p and h_p (2.1) are expanded, in a Taylor series about the point $(e_1^k, e_2^k, \dots, e_N^k, f_1^k, f_2^k, \dots, f_N^k)$, and then proceed to the next step as in the first iteration.

Δ^{kP} and Δ^{kQ} become smaller and smaller as the number of iterations, "k", increase. When these increments become less than a specified tolerance, the process may be stopped and e_i^k , f_i^k , considered as the solution. A flow chart of the method can be found in appendix 3.

It could be easily shown that,

$$J^I_{ik} = e_i G_{ik} - f_i B_{ik} \quad i \neq k$$

$$J^{II}_{ik} = e_i B_{ik} + f_i G_{ik} \quad i \neq k$$

$$J^{III}_{ik} = J^{II}_{ik}$$

$$J^{IV}_{ik} = - J^I_{ik}$$

$$J^I_{ii} = e_i G_{ii} - f_i B_{ii} + \sum_k (e_k G_{ik} + f_k B_{ik})$$

$$J^{II}_{ii} = e_i B_{ii} + f_i G_{ii} + \sum_k (f_k G_{ik} - e_k B_{ik})$$

$$J^{III}_{ii} = e_i B_{ii} + f_i G_{ii} - \sum_k (f_k G_{ik} - e_k B_{ik})$$

$$J^{IV}_{ii} = f_i B_{ii} - e_i G_{ii} + \sum_k (e_k G_{ik} + f_k B_{ik})$$

so the linearized system to be solved would be (2.7),

$$\begin{aligned} \Delta P_p = & \sum_q \{ \Delta e_q (e_p G_{pq} - f_p B_{pq}) + \Delta f_q (e_p B_{pq} + f_p G_{pq}) \} + \\ & + \Delta e_p \sum_q (e_q G_{pq} + f_q B_{pq}) + \Delta f_p \sum_q (f_q G_{pq} - e_q B_{pq}) \end{aligned}$$

$$\begin{aligned} \Delta Q_p = & \sum_q \{ \Delta e_q (e_p B_{pq} + f_p G_{pq}) + \Delta f_q (f_p B_{pq} - e_p G_{pq}) \} + \\ & + \Delta e_p \sum_q (e_q B_{pq} - f_q G_{pq}) + \Delta f_p \sum_q (e_q G_{pq} + f_q B_{pq}) \end{aligned}$$

$$p \neq 1, 2, \dots, N \quad (p \neq S) \quad (2.7)$$

Section 5.5 of P. Henrici's book on numerical methods (10), gives a fair insight into the solution of a set of nonlinear equations by the Newton-Raphson method. For further details, section 3.3 of Isaacson and Kellers' book (11) can be consulted.

2.2 NUMERICAL PROPERTIES

It is proven in section 3.3.2 of reference (11), that if

the following set of conditions are satisfied, the Newton-Raphson method converges to a solution.

Calling J the jacobian and E^0, F^0 the initial iterate of the voltage, it can be proven that,

- if $J^{-1}(E^0, F^0)$ exists and its norm is bounded, "a" being the bound, and
- if the norm of the first set of increments $\Delta^1 e_i, \Delta^1 f_i$ has "b" as a bound for its norm, and
- if for all the E, F that lie in the $2b$ -sphere about E^0, F^0 any of the functions (2.1) $g_p(E, F)$ or $h_p(E, F)$ have continuous second order derivatives with respect to any pair of components of E, F , which satisfy:

$$\sum_k \left| \frac{\delta^2 g_p(E, F)}{\delta e_j \delta e_k} \right| \leq \frac{c}{2xN} \quad \text{and}$$

- if $axbxc \leq \frac{1}{2}$ then,

all Newton-Raphson iterates are within the $2b$ -sphere about E^0, F^0 , and these iterates converge towards a vector which satisfies the equations (1.5).

Furthermore, if the function (2.1) g_p and h_p have two derivatives and the jacobian is non-singular at the solution, the convergence of Newton-Raphson's method is quadratic, which means that the error at the "k+1" iteration is less than the square of the error at the "k" iteration multiplied by a bounded constant.

2.3 ANALYSIS OF COMPUTATION TIME

For analysis of the performance of various methods a number of test systems have been chosen. The IEEE 14, 30, and 57 bus test systems, along with other test systems with higher number of buses, were put forward by the American Electric Power Conference for investigators to use in their research concerned with electrical power systems.

These test systems have the average characteristics of existing power systems as far as electric stability and numerical properties are concerned.

The data for the IEEE 14, 30, and 57 bus test systems can be found in reference (9). Applying the Newton-Raphson method to these three cases, it is found that 3 complete iterations and the start of a fourth iteration are needed.

The iterations have been split into three parts as far as time measuring is concerned. These subdivisions are:

- Calculation of new currents and powers, and checking of the differences between the computed powers according to the iterated voltages, and the scheduled powers.
- Calculation of the new terms of the jacobian and the new set of differences in power.
- Solution of the linear system and correction of the voltages with the set of differences in voltage obtained from the linear system.

The times used are those corresponding to the program shown in appendix 3. The times are average times because the length

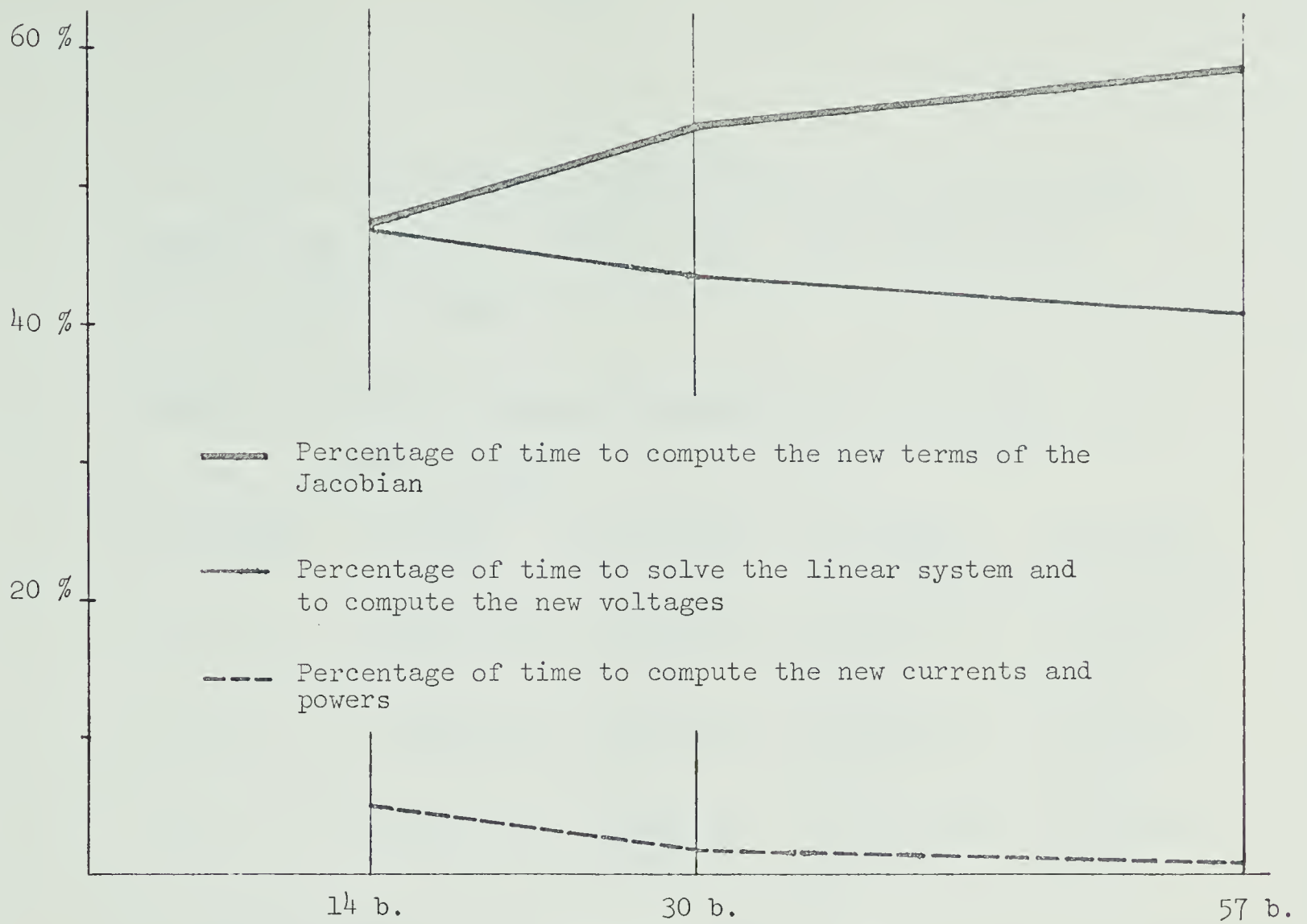


Figure 2.1 PERCENTAGE OF TIME FOR STAGES OF COMPUTATION VS. NUMBER OF BUSES

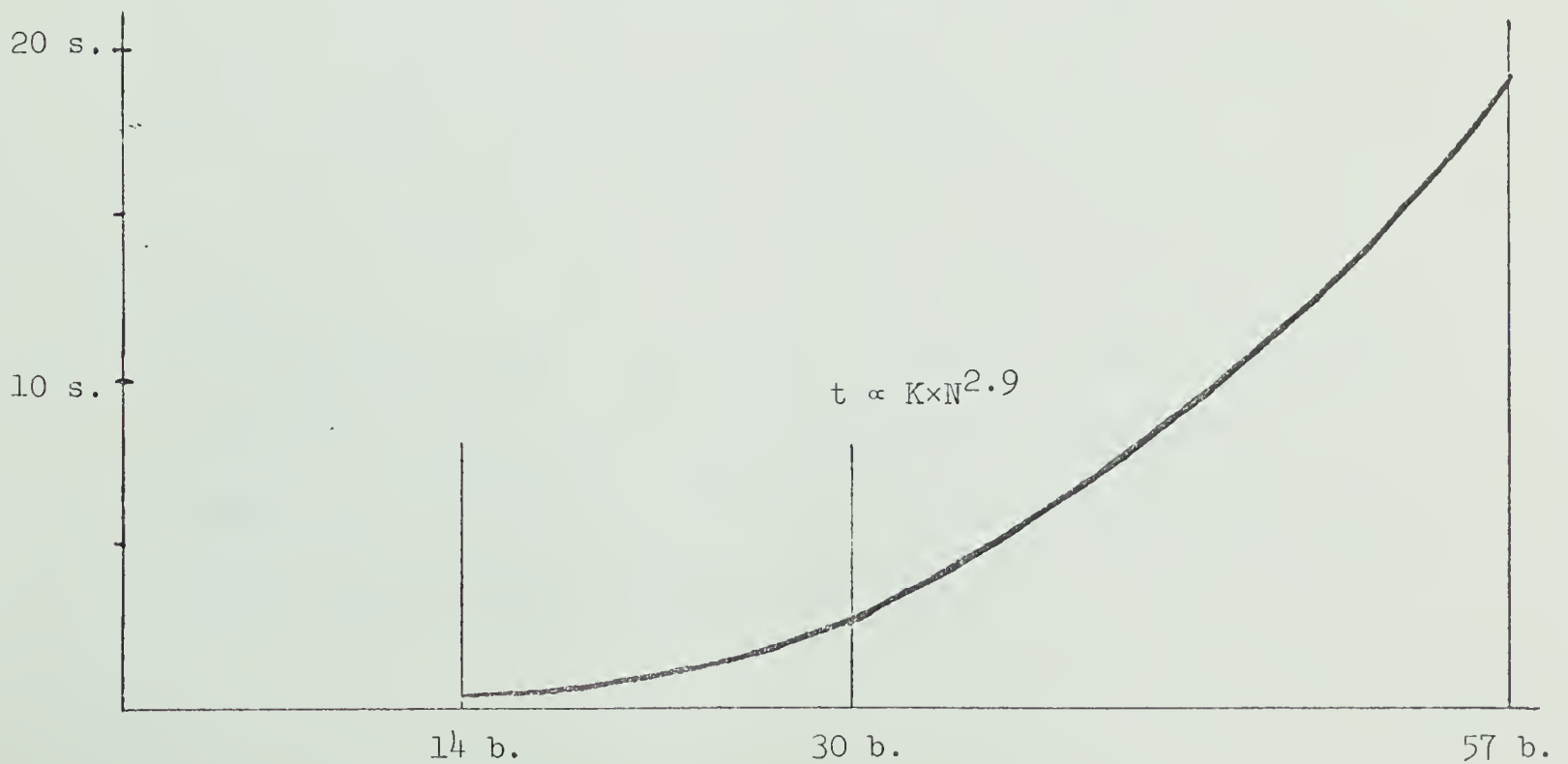


Figure 2.2 TIME PER ITERATION VS. NUMBER OF BUSES

of the same problem is not a constant, due to the fact that the computer used utilizes different modes. The unified scale times are the following:

TABLE 2.1 TIMES FOR STAGES OF COMPUTATION IN THE NEWTON-RAPHSON METHOD

n° of buses	time per iteration	new currents and powers	new terms in jacobian	linear sys. and new volts.
14	0.3254 sec.	0.0166 sec. (5.1 %)	0.1548 sec. (47.6%)	0.1540 sec. (47.3%)
30	2.8653 sec.	0.0526 sec. (1.8%)	1.5604 sec. (54.5%)	1.2523 sec. (43.7%)
57	19.2078 sec.	0.1502 sec. (0.08%)	11.2905 sec. (58.7%)	7.7670 sec. (41.2%)

In brackets are the percentages with respect to the total time per iteration. Graphs of the times and percentages versus the number of buses are given in figures 2.1 and 2.2.

The equation that approximately matches the shape of the curve in figure 2.2, is $t = KxN^{2.9}$, where t stands for time, K is a constant, and N stands for the number of buses.

Most of the time of each iteration, is involved in the calculation of the terms of the jacobian and in the solution of the linearized system. Any savings in time by other methods must replace these parts of the computation by more efficient algorithms. Elimination of the calculation of the jacobian would appear to be the most promising improvement.

III. THE PROPOSED METHOD

3.1 AN ITERATIVE METHOD FOR SOLVING A SET OF NONLINEAR QUADRATIC EQUATIONS

H.S. Wall (13), developed a modification of Newton's method for approximating the roots of a function $f(x)$.

$x_i (i = 0, 1, 2, \dots)$, being the first estimation and the successive iterates of a root, the modification consists in finding the parabola through $x_i, f(x_i)$, having the same first and second derivatives as $f(x)$. The intersection of the parabola through $x_i, f(x_i)$, with the x axis, is the new iterate x_{i+1} . The Newton-Raphson method finds new iterates by means of the tangents to $f(x)$ through the points $x_i, f(x_i)$.

The iterative technique implemented in this thesis finds a parabola through $x_0, f(x_0)$ having the same first and second derivative as $f(x)$ at x_0 , and takes as a new iterate the intersection of this parabola with the x axis. The new iterates are used for improving the coefficient of the second derivative. As this coefficient is unknown, it is taken as zero during the first iteration, which makes the first iteration equivalent to an iteration of the Newton-Raphson method.

The new method is different from Wall's method in that all the parabolas pass through the initial estimate $x_0, f(x_0)$, whereas in the method by Wall the parabola at each iteration pass through the point $x_i, f(x_i)$. Moreover, the approximation used for the coeffi-

cient is different for each method.

The new iterative method is suitable for finding the roots of a quadratic function, because such functions have no third derivative, and the new method uses only the first and the second derivative.

The equations that describe the load-flow problem are quadratic with respect to the bus voltages, which are the unknowns. Therefore, a solution can be expected from the implementation of the proposed method to the load-flow problem.

The new method has some similarities with the Newton-Raphson method. However, due to the nature of the equations and iterations of the new method, its numerical properties are more difficult to prove than those of Newton-Raphson's. The complete numerical features of the new method for one-dimensional cases, are given in section 3.1.1 . The comparison with the one-dimensional case for the Newton-Raphson method, is given in section 3.1.3 .

A sufficient condition for convergence is the multidimensional case is given in section 3.3 . This convergence condition is the application of the general convergence condition for the iterative methods of solution of systems of nonlinear equations.

In section 2.2 are the requirements under which these conditions are fulfilled by the Newton-Raphson method. Proofs can be found in the numerical analysis references (10) and (11). As for the new method in the multidimensional case, the requirements under which the general convergence condition is fulfilled, have not been found theoretically,

because of the complexity of the equations. However, the experience with the one-dimensional case, and the arguments presented in section 3.3 make one believe that, as for the one-dimensional case, the requirements are the same as for the Newton-Raphson method. The IEEE 14, 30 and 57 bus test systems show that, when the proposed method is used, the convergence follows the same rule as the one-dimensional case.

3.1.1 THE NEW ITERATIVE METHOD IN THE ONE-DIMENSIONAL CASE. Consider a quadratic function $q(x)$, from which roots are to be found by an iterative method. If the value x_0 is taken as a first assumption of the root x_* , and $q(x)$ is expanded in a complete Taylor series expansion about x_0 , $q(x_*)$ becomes (3.1a)

$$q(x_*) = 0 = q(x_0) + (x_* - x_0)q'(x_0) + \frac{1}{2}(x_* - x_0)^2q''(x_0) \quad (3.1a)$$

The expression (3.1a) is exact since there are no more derivatives beyond the second, because $q(x)$ is quadratic.

If the nature of $q(x)$ is known, there is an easier way to find the exact development about x_0 . Let us assume that $q(x)$ is given by (3.1b),

$$q(x_*) = 0 = (x_* - r_1)(x_* - r_2) \quad (3.1b)$$

where r_1 and r_2 are the roots of $q(x)$. Equation (3.1b) can be written in an equivalent way.

$$q(x_*) = q(x_* - x_0 + x_0) = (x_* - x_0 + x_0 - r_1)(x_* - x_0 + x_0 - r_2) \quad (3.1c)$$

which is equivalent to (3.1d)

$$q(x_*) = (x_0 - r_1)(x_0 - r_2) + \{(x_0 - r_1) + (x_0 - r_2)\}(x_* - x_0) + (x_* - x_0)^2 \quad (3.1d)$$

and since $q(x)$ is as in (3.1b),

$$q(x_0) = (x_0 - r_1)(x_0 - r_2) \quad (3.1e)$$

$$q'(x_0) = (x_0 - r_1) + (x_0 - r_2) \quad (3.1f)$$

$$\frac{1}{2}q''(x_0) = 1 \quad (3.1g)$$

if (3.1e), (3.1f), and (3.1g) are substituted in (3.1d), (3.1a) is obtained. This was to be expected because all the exact developments of $q(x)$ have to be equivalent.

The way of developing $q(x)$ in (3.1d) is used in the derivation of the formulas for the application of the proposed method to the load-flow problem, because it yields the final equations in less steps than does the Taylor series expansion. This method is implemented in Appendix 1 as well.

Equation (3.1a) can be rewritten in a different way (3.1h), from which an iterative procedure for finding x_* can be derived,

$$(x_* - x_0)\{q'(x_0) + \frac{q''(x_0)}{2}(x_* - x_0)\} = -q(x_0) \quad (3.1h)$$

and the iterative technique would be (3.2a),

$$x_{i+1} = x_0 - \frac{q(x_0)}{q'(x_0) + \frac{q''(x_0)}{2} (x_i - x_0)} \quad (3.2a)$$

For the solution value x_* , this gives (3.2b)

$$x_* = x_0 - \frac{q(x_0)}{q'(x_0) + \frac{q''(x_0)}{2} (x_* - x_0)} \quad (3.2b)$$

The convergence requirements for an iterative formula such as (3.3a), are the following:

Given the iterative sequence,

$$x_{i+1} = g(x_i) \quad i = 0, 1, 2, \dots \quad (3.3a)$$

whose initial estimate will be called x_0 , then,

- if 1) $g(x)$ satisfies the Lipschitz condition

$$|g(x) - g(x')| \leq \lambda |x - x'|, \quad 0 \leq \lambda < 1 \quad (3.3b)$$

for all the values x, x' in the closed interval

$I \equiv [x_0 - \rho, x_0 + \rho]$, where λ is the Lipschitz constant; and

- if 2) the initial estimate x_0 is such that

$$|x_0 - g(x_0)| \leq (1 - \lambda) \rho \quad (3.3c)$$

then, a) all the iterates defined by $x_{i+1} = g(x_i)$ lie within the

interval I ,

b) the iterates x_i , converge to some point

$$\lim_{i \rightarrow \infty} x_i = \alpha \quad (3.3d)$$

which is a root, i.e.: $\alpha = g(\alpha)$; and

c) α is the only root in the interval I

Furthermore, it can be shown that:

- if 3) for any x within the interval I

$$|g'(x)| \leq \lambda \quad (3.3e)$$

and

- if 4) the initial estimate x_0 satisfies the condition 2),

(3.3c), then,

Conclusions a), b), and c) are valid.

Assuming that x_0 complies with 4), the convergence of the new method can be checked by means of the condition 3), equation (3.3e).

During the first iteration, x_0 is a variable, but during the other iterations, x_0 is a constant parameter. Therefore, there will be a different function $g'(x_i)$ for the first and the other iterations. Deriving (3.2a) with respect to x_0 , we get $g'(x_0)$. Thus,

$$g'(x_0) = 1 - \frac{q'(x_0)^2 - q(x_0)q''(x_0)}{q'(x_0)^2} = \frac{q(x_0)q''(x_0)}{q'(x_0)^2} \quad (3.4a)$$

and thus when x_0 belongs to the interval I, the initial value must satisfy,

$$|g'(x_0)| = \frac{|q(x_0)q''(x_0)|}{q'(x_0)^2} \leq \lambda \quad (3.4b)$$

For the iterations beyond the first, x_0 is regarded as a constant. Taking the derivation of the right hand side of (3.2a) with respect to x_i gives (3.4c)

$$|g'(x_i)| = \frac{|q(x_0) \frac{q''(x_0)}{2}|}{\{q'(x_0) + \frac{q''(x_0)}{2}(x_i - x_0)\}^2} \leq \lambda \quad (3.4c)$$

By subtracting (3.2a) from (3.2b), i.e.: x_{i+1} from x_* an expression for the error at the $(i+1)$ iteration is obtained.

$$x_* - x_{i+1} = \varepsilon_{i+1} = \frac{q(x_0) \frac{q''(x_0)}{2} \varepsilon_i}{\{q'(x_0) + \frac{q''(x_0)}{2}(x_i - x_0)\}\{q'(x_0) + \frac{q''(x_0)}{2}(x_* - x_0)\}} \quad (3.5a)$$

However, for the first iteration, the expression is different. Considering (3.1a), if the expression is divided by $q'(x_0)$, this gives,

$$0 = \frac{q(x_0)}{q'(x_0)} + x_* - x_0 + \frac{q''(x_0)}{2q'(x_0)}(x_* - x_0)^2 \quad (3.5b)$$

but according to (3.2a) applied for x_1 ,

$$x_1 = x_0 - \frac{q(x_0)}{q'(x_0)} \quad (3.5c)$$

From (3.5b) and (3.5c), and bearing in mind that $\varepsilon_1 = x_* - x_1$ and $\varepsilon_0 = x_* - x_0$, the error at the end of the first iteration is (3.5d),

$$\epsilon_1 = - \frac{q''(x_0)}{2q'(x_0)} \epsilon_0^2 \quad (3.5d)$$

A new expression can be found for ϵ_{i+1} . Multiplying the numerator and denominator of (3.5a) by $\epsilon_0 = (x_* - x_0)$, and recalling (3.1h), the expression (3.5e) can be written,

$$\epsilon_{i+1} = - \frac{q(x_0) \frac{q''(x_0)}{2} \epsilon_0 \epsilon_i}{q(x_0) \{q'(x_0) + \frac{q''(x_0)}{2} (x_i - x_0)\}} \quad (3.5e)$$

Dividing the numerator and the denominator of (3.5e) by $q(x_0)q'(x_0)$, and noting that $x_i - x_0 = \epsilon_0 - \epsilon_i$, this gives,

$$\epsilon_{i+1} = \frac{- \frac{q''(x_0)}{2q'(x_0)} \epsilon_0 \epsilon_i}{1 + \frac{q''(x_0)}{2q'(x_0)} (\epsilon_0 - \epsilon_i)} \quad (3.5f)$$

and taking the expression (3.5d) into account,

$$\epsilon_{i+1} = \frac{\frac{\epsilon_1}{\epsilon_0}}{1 - \frac{\epsilon_1}{\epsilon_0} + \frac{\epsilon_1}{\epsilon_0} \frac{\epsilon_i}{\epsilon_0}} \epsilon_i \quad (3.5g)$$

For a convergent problem, when the iterates are close to the solution, ϵ_i/ϵ_0 will be close to zero, and the term $\epsilon_1\epsilon_i/\epsilon_0\epsilon_0$, in the denominator of (3.5g), can be dropped. The successive errors in the neighbourhood of the solution will satisfy the approximate formula (3.5h)

$$\epsilon_{i+1}^s = \frac{\frac{\epsilon_1}{\epsilon_0}}{1 - \frac{\epsilon_1}{\epsilon_0}} \epsilon_i^s \quad (3.5h)$$

where the subscript "s" in the error, indicates that these errors are small in comparison with the initial error, because the iterates are close to the solution.

By comparing (3.5d), (3.5g), and (3.5h), it can be concluded that the convergence goes from quadratic at the first iteration to linear at the end of the process.

3.1.2 THE NEWTON-RAPHSON METHOD IN THE ONE-DIMENSIONAL CASE. Considering the development (3.6),

$$q(x_*) = 0 = q(x_i) + (x_* - x_i)q'(x_i) + \frac{1}{2} (x_* - x_i)^2 q''(x_i) \quad (3.6)$$

and dropping the last right hand side term of (3.6), the Newton-Raphson's iterative procedure is obtained,

$$x_{i+1} = x_i - \frac{q(x_i)}{q'(x_i)} \quad (3.7)$$

Recalling the iterative formula (3.3a) and the convergence condition (3.3e), it can be written for the Newton-Raphson method,

$$|g'(x_i)| = \frac{|q(x_i)q''(x_i)|}{q'(x_i)^2} \leq \lambda \quad (3.8)$$

Dividing (3.6) by $q'(x)$, the error formula (3.9) is obtained,

$$\varepsilon_{i+1} = - \frac{q''(x_i)}{2q'(x_i)} \varepsilon_i^2 \quad (3.9)$$

which means that the convergence is always quadratic.

3.1.3 COMPARISON OF THE CONVERGENCE OF THE NEW METHOD WITH A ONE-DIMENSIONAL QUADRATIC SYSTEM, WITH THAT OF NEWTON-RAPHSON'S. For the Newton-Raphson method there always exists an interval where the convergence condition (3.8) is fullfilled, provided that $g'(x)$ is a continous function, because $g'(x_*) = 0$, as $q(x_*) = 0$.

Since the formuals (3.5d) and (3.9) are equivalent, it can be concluded that the convergence of both methods are exactly alike during the first iteration.

For the convergence of any two succesive iterations of the Newton-Raphson method, the following condition, derived from (3.9), must be satisfied for the initial estimate x_0 , and all the iterates x_i ,

$$\left| \frac{\epsilon_{i+1}}{\epsilon_i} \right| = \left| - \frac{q''(x_i)}{2q'(x_i)} \epsilon_i \right| \leq 1 \quad i = 0, 1, 2, \dots \quad (3.10)$$

Similarly for the proposed method, if any $\epsilon_{i+1}/\epsilon_i$ has to be equal or less than one, from (3.5g), it can be derived that,

$$-1 \leq \frac{\frac{\epsilon_1}{\epsilon_0}}{1 - \frac{\epsilon_1}{\epsilon_0} + \frac{\epsilon_1}{\epsilon_0} \frac{\epsilon_i}{\epsilon_0}} \leq 1 \quad (3.11a)$$

The condition (3.11a) must be satisfied to guarantee the convergence of the method.

The term ϵ_i/ϵ_0 for every "i", can be put in terms of ϵ_1/ϵ_0 by means of the equation (3.5g). The expression (3.11a) with ϵ_i/ϵ_0 in terms of ϵ_1/ϵ_0 , becomes more and more complicated as "i" increases. Although a general proof for any "i" has not been derived, it has been found that (3.11a) is satisfied for the values of "i", 1,2,3, and 4, provided that $|\epsilon_1/\epsilon_0|$ is less than one.

3.2 DEVELOPMENT OF THE METHOD

The proposed method can be put in the same classification as the Newton-Raphson method; both are direct methods, because the solution of a system of nonlinear equations gives a set of voltage corrections. As for the Newton-Raphson method, the proposed method is based on the set of nonlinear equations (1.5), thus the bus admittance form is used.

If the system of nonlinear equations (1.5) is examined,

$$\begin{aligned} P_p &= \sum_q \{e_p(e_q G_{pq} + f_q B_{pq}) + f_p(f_q G_{pq} - e_q B_{pq})\} \\ Q_p &= \sum_q \{f_p(e_q G_{pq} + f_q B_{pq}) - e_p(f_q G_{pq} - e_q B_{pq})\} \end{aligned} \quad (1.5)$$

it can be seen that each equation consists of a sum of terms, each being a quadratic expression of the bus voltages. The general form of the terms is always one of the following: $G_{ij}e_i f_j$, $G_{ij}e_i f_j$, $G_{ij}f_i f_j$, $B_{ij}e_i e_j$, $B_{ij}e_i f_j$, $B_{ij}f_i f_j$.

The proposed method is based on an iterative method of solution of a system of quadratic nonlinear equations that is equivalent to the system (1.5). The way to obtain the equivalent system and its solution is analogous to that of appendix 1, where a two dimensional example of the procedure is shown.

Let the set $e^*_i, f^*_i, (i = 1, 2, \dots, N)$, be the solution of (1.5), and let e^o_i, f^o_i , be a set of values which are not the solution. Let $\Delta e^o_i, \Delta f^o_i$ be the set of increments, such that they bring e^o_i, f^o_i , to the exact solution of (1.5), e^*_i, f^*_i ,

$$\begin{aligned} e^*_i &= e^o_i + \Delta e^o_i \\ f^*_i &= f^o_i + \Delta f^o_i \end{aligned} \quad (3.13)$$

substituting (3.13) in (1.5),

$$\begin{aligned} P_p &= \sum_q \{ (e^o_p + \Delta e^o_p) [(e^o_q + \Delta e^o_q) G_{pq} + (f^o_q + \Delta f^o_q) B_{pq}] + \\ &\quad + (f^o_p + \Delta f^o_p) [(f^o_q + \Delta f^o_q) G_{pq} - (e^o_q + \Delta e^o_q) B_{pq}] \} \\ Q_p &= \sum_q \{ (f^o_p + \Delta f^o_p) [(e^o_q + \Delta e^o_q) G_{pq} + (f^o_q + \Delta f^o_q) B_{pq}] - \\ &\quad - (e^o_p + \Delta e^o_p) [(f^o_q + \Delta f^o_q) G_{pq} - (e^o_q + \Delta e^o_q) B_{pq}] \} \end{aligned} \quad (3.14a)$$

calling,

$$\begin{aligned} P^o_p &= \sum_q \{ e^o_p (e^o_q G_{pq} + f^o_q B_{pq}) + f^o_p (f^o_q G_{pq} - e^o_q B_{pq}) \} \\ Q^o_p &= \sum_q \{ f^o_p (e^o_q G_{pq} + f^o_q B_{pq}) - e^o_p (f^o_q G_{pq} - e^o_q B_{pq}) \} \end{aligned} \quad (3.14b)$$

and $P_p = P_p - P^o_p$, $\Delta Q_p = Q_p - Q^o_p$, then

$$\begin{aligned} P_p - P^o_p &= \sum_q \{ e^o_p (\Delta e^o_q G_{pq} + \Delta f^o_q B_{pq}) + f^o_p (\Delta f^o_q G_{pq} - \Delta e^o_q B_{pq}) \} + \\ &\quad + \Delta e^o_p \sum_q \{ (e^o_q + \Delta e^o_q) G_{pq} + (f^o_q + \Delta f^o_q) B_{pq} \} + \\ &\quad + f^o_p \sum_q \{ (f^o_q + \Delta f^o_q) G_{pq} - (e^o_q + \Delta e^o_q) B_{pq} \} \end{aligned} \quad (3.14c)$$

$$\begin{aligned} Q_p - Q^o_p &= \sum_q \{ f^o_p (\Delta e^o_q G_{pq} + \Delta f^o_q B_{pq}) - e^o_p (\Delta f^o_q G_{pq} - \Delta e^o_q B_{pq}) \} + \\ &\quad + \Delta e^o_p \sum_q \{ (e^o_q + \Delta e^o_q) B_{pq} - (f^o_q + \Delta f^o_q) G_{pq} \} + \\ &\quad + \Delta f^o_p \sum_q \{ (e^o_q + \Delta e^o_q) G_{pq} + (f^o_q + \Delta f^o_q) B_{pq} \} \end{aligned}$$

which is equivalent to

$$\begin{aligned}\Delta P_p &= \sum_q \{ \Delta e^o_q (e^o_p G_{pq} - f^o_p B_{pq}) + \Delta f^o_q (e^o_p B_{pq} + f^o_p G_{pq}) \} \\ &+ \Delta e^o_p \sum_q \{ (e^o_q + \Delta e^o_q) G_{pq} + (f^o_q + \Delta f^o_q) B_{pq} \} \\ &+ \Delta f^o_p \sum_q \{ (f^o_q + \Delta f^o_q) G_{pq} - (e^o_q + \Delta e^o_q) B_{pq} \}\end{aligned}\tag{3.14d}$$

$$\begin{aligned}\Delta Q_p &= \sum_q \{ \Delta e^o_q (e^o_p B_{pq} + f^o_p G_{pq}) + f^o_q (f^o_q B_{pq} - e^o_p G_{pq}) \} \\ &+ \Delta e^o_p \sum_q \{ (e^o_q + \Delta e^o_q) B_{pq} - (f^o_q + \Delta f^o_q) G_{pq} \} \\ &+ \Delta f^o_p \sum_q \{ (e^o_q + \Delta e^o_q) G_{pq} + (f^o_q + \Delta f^o_q) B_{pq} \}\end{aligned}$$

The system (3.14d) is equivalent to system (1.5). This implies that if the system of nonlinear equations (3.14d) is solved, the solution of the load-flow problem will be known. The unknowns of the system (3.14d) are the set of increments $\Delta e^o_i, \Delta f^o_i$.

It is possible to arrange the equations (3.14d) in such a way that the subscripts of the left hand side constants are arranged in increasing order; first for ΔP_i , and afterwards for ΔQ_i , as in $\Delta P_1, \Delta P_2, \dots, \Delta P_N, \Delta Q_1, \Delta Q_2, \dots, \Delta Q_N$. It is also possible to arrange the right hand side terms, so that the unknowns have their subscripts in increasing order. At first for Δe_i , and afterwards for Δf_i , as in $\Delta e_1, \Delta e_2, \dots, \Delta e_N, \Delta f_1, \Delta f_2, \dots, \Delta f_N$.

Assuming that the ordering of equations and terms described above has been done, and writing (3.14d) in matrix form, we can observe that the coefficients are all constant except those in the

diagonals of the four similar submatrices into which the matrix of coefficients can be subdivided. The coefficients in the four diagonals have a constant part, which belong to the sum which is the first term in the right hand side of the equations (3.14d). And they have a variable part, which depends on the value of the solution $\Delta e^0_i, \Delta f^0_i$. This part is called the variable part because it will change value during the successive iterations, as we find better values for the solution $\Delta e^0_i, \Delta f^0_i$.

The iterative method of solution that can be applied is the same as that in Appendix 1. The nonlinear system (3.14d) is linearized by giving an approximate value to the unknown part of the variable part of the diagonal coefficients. For instance, in the diagonal coefficient for Δe^0_p , the variable part is,

$$\sum_q \{ (e^0_q + \Delta e^0_q) G_{pq} + (f^0_q + \Delta f^0_q) B_{pq} \} \quad (3.15a)$$

since the solution of (3.14d) is part of the variable part of the coefficient, and the solution is not known, a value for Δe^0_i , and Δf^0_i , will have to be assumed. In the first iteration, these values will be taken as zero. In other words, Δe^0_i , and Δf^0_i will be dropped before e^0_i , and f^0_i . The variable part of the diagonal coefficient of Δe^0_p will be then,

$$\sum_q \{ (e^0_q G_{pq} + f^0_q B_{pq}) \} \quad (3.15b)$$

which is constant. Doing the same with all the diagonal coefficients, a linearized system results. Its solution will be the

set of values $\Delta e^1_i, \Delta f^1_i$. These values can be substituted in (3.15a). The variable part of the diagonal coefficient of Δe^0_p , is then,

$$\sum_q \{ (e^0_q + \Delta e^1_q) G_{pq} + (f^0_q + \Delta f^1_q) B_{pq} \} \quad (3.15c)$$

which is again a constant. Doing the same with all the diagonal elements, a new linearized system of equations will be obtained. Once solved, it will give a new set of solutions $\Delta e^2_i, \Delta f^2_i$.

It can be expected that $\Delta e^2_i, \Delta f^2_i$ is a better approximation to $\Delta e^0_i, \Delta f^0_i$, than $\Delta e^1_i, \Delta f^1_i$, because (3.15c) is closer to (3.15a) than (3.15b) is. The same process can be applied again, i.e.: substitute $\Delta e^2_i, \Delta f^2_i$ in (3.15a) and obtain a new and better set $\Delta e^3_i, \Delta f^3_i$.

When, at the first iteration, $\Delta e^0_p, \Delta f^0_p$ are dropped before e^0_p, f^0_p , equation (3.14d) becomes,

$$\begin{aligned} \Delta P_p = & \sum_q \{ \Delta e_q (e^0_p G_{pq} - f^0_p B_{pq}) + \Delta f_q (e^0_p B_{pq} + f^0_p G_{pq}) \} + \\ & + \Delta e_p \sum_q \{ e^0_q G_{pq} + f^0_q B_{pq} \} + \\ & + \Delta f_p \sum_q \{ f^0_q G_{pq} - e^0_q B_{pq} \} \end{aligned} \quad (3.16a)$$

$$\begin{aligned} \Delta Q_p = & \sum_q \{ \Delta e_q (e^0_p B_{pq} + f^0_p G_{pq}) + \Delta f_q (f^0_p B_{pq} - e^0_p G_{pq}) \} + \\ & + \Delta e_p \sum_q \{ e^0_q B_{pq} - f^0_q G_{pq} \} + \Delta f_p \sum_q \{ e^0_q G_{pq} + f^0_q B_{pq} \} \end{aligned}$$

(3.16a) would give the set $\Delta e^1_i, \Delta f^1_i$ as a solution. In general at the (k+1) iteration the linearized system (3.16b) would be solved

$$\begin{aligned}
 \Delta P_p = & \sum_q \{ \Delta e_q (e^o_p G_{pq} - f^o_p B_{pq}) + \Delta f_q (e^o_p B_{pq} + f^o_p G_{pq}) \} + \\
 & + \Delta e_p \sum_q \{ (e^o_q + \Delta e^k_q) G_{pq} + (f^o_q + \Delta f^k_q) B_{pq} \} + \\
 & + \Delta f_p \sum_q \{ (f^o_q + \Delta f^k_q) G_{pq} - (e^o_q + \Delta e^k_q) B_{pq} \}
 \end{aligned} \tag{3.16b}$$

$$\begin{aligned}
 \Delta Q_p = & \sum_q \{ \Delta e_q (e^o_p B_{pq} + f^o_p G_{pq}) + \Delta f_q (f^o_p B_{pq} - e^o_p G_{pq}) \} + \\
 & + \Delta e_p \sum_q \{ (e^o_q + \Delta e^k_q) B_{pq} - (f^o_q + \Delta f^k_q) G_{pq} \} + \\
 & + \Delta f_p \sum_q \{ (e^o_q + \Delta e^k_q) G_{pq} + (f^o_q + \Delta f^k_q) B_{pq} \}
 \end{aligned}$$

which would yield the set of values $\Delta e^{k+1}_i, \Delta f^{k+1}_i$, as solution.

3.3 THE BASES FOR CONVERGENCE

Observing (2.7), and (3.16a) in the former section, it can be seen that they are the same. (2.7) is the general linearized system which is solved at each iteration of the Newton-Raphson method, and (3.16a) is the linearized system which is solved at the first iteration of the proposed method. From this it can be concluded that the first iteration of the proposed method is exactly the same as that of the Newton-Raphson method, provided that the same starting value for the voltages is taken.

Therefore, as far as the first iteration is concerned, the same reasoning can be applied as for the numerical properties of the Newton-Raphson method, this means that if the set of conditions specified in section 2.2 are satisfied for the initial guess of the voltages, then the new voltages, which result from correcting

the initial guess with the increments that result from the first iteration are always closer to the solution than the voltages of the first guess are. Moreover, in the case of the proposed method, it means as well that all Δe^1_i , Δf^1_i , point towards the solution.

As it has been shown in section 3.2, the variable parts of the diagonal coefficients, once the values of Δe^1_i , Δf^1_i , have been substituted, are closer to the true values of the diagonal coefficients than that of the first iteration. The non-diagonal coefficients in the proposed method are unchanged from the first until the last iteration.

As for the one dimensional case in section 3.1, the results in the second iteration Δe^2_i , Δf^2_i , will be closer to the solution Δe^0_i , Δf^0_i , than the set Δe^1_i , Δf^1_i . This is so because the linearized system (3.16b) for iteration 2 is closer to the exact system (3.14d) than the linearized system (3.16a) for the first iteration is.

In other words, since the non-diagonal coefficients are the same for the exact system and for the approximate system, it can be expected that if the value of the diagonal coefficients tend to their true value, the solutions at each iteration have to tend towards the true value of the solution of the linear system (3.14d).

Calling ΔS the vector of differences of powers, T the voltage vector, L the matrix of coefficients expressing the linear part of the system of equations (3.14b), and D the matrix of nonlinear diagonal elements, (3.14b) can be written in the simplified way (3.17a),

$$\Delta S = [L + D(T)](T - T_0) \quad (3.17a)$$

from which, the iterative expression (3.17b) is obtained

$$T_{i+1} = T_0 + [L + D(T_i)]^{-1} \Delta S \quad (3.17b)$$

Applying the condition (3.3b), a sufficient convergence condition (3.18), can be written

$$\left\| \frac{\delta}{\delta T_i} [L + D(T_i)]^{-1} \Delta S \right\| \leq 1 \quad (3.18)$$

For the first iteration, the same sufficient convergence condition as for the Newton-Raphson method, can be applied. Therefore, the requirements described in section (2.2) apply for the first iteration of the proposed method.

The experimental results with the IEEE 14, 30, and 57 bus test systems confirm these assertions.

3.4 COMPARISON BETWEEN THE PROPOSED METHOD AND THE NEWTON-RAPHSON METHOD

Although the differences between the Newton-Raphson method and the proposed method have been indicated already, more insight could be obtained if some of the points are stressed, and some of the parts of both methods are put side by side.

For the sake of clarity while the comparison of the two methods is carried out, the increment with a point inside " Δ " will be used to refer to the increments that appear in the proposed method.

Let CIR_p and CII_p be the real and imaginary part of the current

flowing into the system at bus "p". It is known that when the bus voltages are e_i^* , f_i^* , the real and imaginary component of the current at bus "p" are,

$$\begin{aligned} CIR_p^* &= \sum_q (e_q^* G_{pq} + f_q^* B_{pq}) \\ CII_p^* &= \sum_q (f_q^* B_{pq} - e_q^* G_{pq}) \end{aligned} \quad (3.19)$$

observing (2.7) it can be seen that (3.19) could be substituted in there. A new expression of (2.7) could be written,

$$\begin{aligned} \Delta P_p &= \sum_q \{ \Delta e_q (e_p G_{pq} - f_p B_{pq}) + \Delta f_p (e_p B_{pq} + f_p G_{pq}) \} + \\ &+ \Delta e_p CIR_p + \Delta f_p CII_p \\ \Delta Q_p &= \sum_q \{ \Delta e_q (e_p B_{pq} + f_p G_{pq}) + \Delta f_q (f_p B_{pq} - e_p G_{pq}) \} + \\ &+ \Delta f_p CIR_p - \Delta e_p CII_p \end{aligned} \quad (3.20)$$

As in (3.19), e_i^o , f_i^o , being the initial guess voltages, it is possible to define,

$$\begin{aligned} CIR_p^o &= \sum_q (e_q^o G_{pq} + f_q^o B_{pq}) \\ CII_p^o &= \sum_q (f_q^o B_{pq} - e_q^o G_{pq}) \end{aligned} \quad (3.21a)$$

Let DCR_p^o and DCI_p^o be the real and imaginary part of the difference between the current at bus "p" when the voltages at the nodes are the solution voltages, and the current at bus "p" when the voltages are the initial guess,

$$DCR_p^o = \sum_q (\Delta e_q^o G_{pq} + \Delta f_q^o B_{pq})$$

$$CII_p^o = \sum_q (\Delta f_q^o B_{pq} - \Delta e_q^o G_{pq}) \quad (3.21b)$$

The expressions (3.21a) and (3.21b) can be substituted in (3.14d) to get

$$\begin{aligned} \Delta P_p &= \sum_q \{ \Delta e_q^o (e_p^o G_{pq} - f_p^o B_{pq}) + \Delta f_q^o (e_p^o B_{pq} + f_p^o G_{pq}) \} + \\ &\quad + \Delta e_p^o (CIR_p^o + DCR_p^o) + \Delta f_p^o (CII_p^o + DCI_p^o) \\ \Delta Q_p &= \sum_q \{ \Delta e_q^o (e_p^o B_{pq} + f_p^o G_{pq}) + \Delta f_q^o (f_p^o B_{pq} - e_p^o G_{pq}) \} + \\ &\quad + \Delta f_p^o (CIR_p^o + DCR_p^o) - \Delta e_p^o (CII_p^o + DCI_p^o) \end{aligned} \quad (3.22)$$

(3.22) is an equivalent system to (3.14d). It can be noted that in the proposed method the recalculation of the new diagonal coefficients at each iteration is equivalent to finding a better set of increments of currents DCR_i , DCI_i .

Obviously when at the first iteration Δe_p^o , Δf_p^o , are dropped before e_p^o , f_p^o , it is the same as dropping DCR_p^o , DCI_p^o , before CIR_p^o , CII_p^o .

3.4.1 DIFFERENCE BETWEEN THE CONCEPT OF INCREMENT OF VOLTAGE IN THE NEWTON-RAPHSON METHOD AND IN THE PROPOSED METHOD. The difference between the Newton-Raphson method and the proposed method lies basically in their meaning of the increment of voltage.

In the Newton-Raphson method the increments of voltage Δe_i , Δf_i ,

are the difference between the values of the voltages at two successive iterations. In the proposed method the increments of voltage Δe_i , Δf_i , are the difference between the voltage calculated at the end of the iteration, and the first assumed value of the voltages.

Hence, whereas the Newton-Raphson's increments of voltage become smaller and smaller at each iteration, the proposed method has increments Δe_i , Δf_i , that tend towards Δe_i^0 , Δf_i^0 , which are different from zero.

Comparing (3.20) and (3.22), it can be seen that the variable terms in the variable part of the diagonal coefficients in the Newton-Raphson method are the bus currents calculated at the iterated voltages, and are the same for the proposed method, because,

$$\begin{aligned} CIR_p^0 + DCR_p &= CIR_p \\ CII_p^0 + DCI_p &= CII_p \end{aligned} \tag{3.23}$$

It must be kept in mind though that the constant part of the diagonal coefficients is variable at each iteration in the Newton-Raphson method, but is constant in the proposed method.

One more fact can be observed, ΔP_p , ΔQ_p , are the differences between the scheduled powers and the powers computed with the iterated voltages, since the iterated voltages change at each iteration ΔP_p , ΔQ_p will be different too, and as the iterated voltages tend towards the true voltages, the computed powers will tend to the scheduled powers, so ΔP_p , ΔQ_p , tend to zero.

In the proposed method ΔP_p , ΔQ_p are constant because they are the difference between the scheduled powers and the powers calculated with the first assumed value of the voltages.

3.4.2 CALCULATION OF THE ELEMENTS OF THE JACOBIAN. From an examination of (2.7) and (3.14d), it can be noted that whereas in (3.14d) the non-diagonal coefficients are constant, because they depend on the value of the first estimate of the voltages, the non-diagonal coefficients of the Jacobian, when the Newton-Raphson method is used, depend on the value of the iterated voltages, so they have to be recalculated at each iteration.

This is one of the most important assets of the proposed method. As far as computer time is concerned, the recalculation of the elements of the linearized system is the largest item. This is so because of the large size of the matrix of the linearized system, $2(NB - S) \times 2(NB - S)$, (NB number of buses, S number of slack buses), and because of the logical instructions involved in the handling of the elements of the bus admittance matrix.

If with the proposed method, advantage is to be taken of the fact that the non-diagonal coefficients are the same throughout all the iterations, a non-destructive technique for solving the linear system (3.16b), must be used. If a destructive technique is implemented, extra-storage must be provided to save the constant elements of the matrix.

3.5 OUTSTANDING FEATURES OF THE PROPOSED METHOD

As a brief outline of the characteristics of the proposed method, it could be said that:

- The proposed method is a direct method operating with the bus admittance form.
- The proposed method implements a new way to solve the system of nonlinear equations (1.5), which is suitable for a system of quadratic nonlinear equations.
- In the proposed method the only coefficients that change with the iterated voltage, are the diagonal ones, so they have to be recalculated at each iteration.
- The convergence conditions of the proposed method are different from those of the Newton-Raphson method. The conditions for the one-dimensional case are in section 3.1.3 .
- During the first iteration the rate of convergence of the proposed method is quadratic. The rate of convergence is less than quadratic and decrease as the number of iterations increase. The rate of convergence becomes linear in the neighbourhood of the solution.
- It is very easy to switch from the proposed method to the Newton-Raphson method and back, as far as computer programming is concerned. A flow chart of the proposed method can be found in Appendix 3.

IV. THE PROPOSED METHOD USING TRIANGULATION
WITH PIVOTING FOR SOLVING THE LINEARIZED
SYSTEM AT EACH ITERATION. RESULTS
AND CRITICISM

4.1 NUMERICAL RESULTS OF THE PROPOSED METHOD

In this chapter the performance of the proposed method using triangulation with pivoting is examined. The experimentation done has shown that other ways than the exact solution by triangulation with pivoting are more suitable for the solution of the linearized system (3.14d) at each iteration when using the proposed method.

Nevertheless, since triangulation with pivoting is the usual technique for the Newton-Raphson method, it is worth knowing what the proposed method does with it, and analyzing the results obtained.

4.1.1 EXPLANATIONS ON THE COMPUTER PROGRAM USED. The goal of this chapter is the comparison of the methods using the technique, of solution of a linear system of equations, usually implemented by the Newton-Raphson method, namely, triangulation with pivoting.

In order to take advantage of the constancy of the non-diagonal coefficients of the linearized system, double computer storage must be used for the coefficients because triangulation with pivoting destroys the linear system when it finds its solution.

In the computer program used, the whole bus admittance matrix Y_{bus} is used, and no logic programming complications exist for handling the elements of this matrix. This is pointed out because in other sections

the load-flow program used implements sets of logic instructions aimed at avoiding the storage of the zero elements of the bus admittance matrix.

Whenever a program which measures load-flow solution times has been run, it has been run along with a reference case so that the times can always be compared. It is known that a computer can spend different amounts of time on the same problem, due to the different modes of shared time.

There is, in the load-flow problem shown in Appendix 3, a trial and error technique that takes into account the constraints in the available reactive powers at the voltage controlled buses. However, for measuring the convergence and speed of solution of the load-flow method, the first run of the trial and error technique is enough. This is what has been made throughout this thesis.

4.1.2 TEST SYSTEM USED. The IEEE 14 and 30 bus test systems were used for making a comparison between the Newton-Raphson method and the proposed method. The results have been different as far as time is concerned between the two methods, but similar for each test system.

For the sake of brevity, only the results obtained with the 30 bus test system will be shown.

4.1.3 NUMERICAL RESULTS. The tolerance used in the comparison ranges from 0.0001 to 0.0002 units of power or voltage, and will be indicated in every case. The precision will be called EPS.

As can be seen in Appendix 3 there are two ways to get out of the iterative procedure of the proposed method. One by meeting the precision

in power at the buses, and the other when the changes of voltage at the iteration are less than the required precision, EPS.

It has been observed that, in most of the cases, when the change in voltage is less than EPS, the mismatch in power at the buses with respect to the scheduled powers is not necessarily less than EPS. An error can thus be found at the end of the process. It could be eliminated by requiring a higher precision in voltage change to be met, but there are some reasons for not doing so. A further discussion about these points can be found in the section where there are the comments on the advantages and disadvantages of the proposed method. In this section it will be seen as well that the rate of convergence decreases as the solution is approached.

The performance of the Newton-Raphson method and the proposed method are shown in Table (4.1).

TABLE 4.1 PERFORMANCE OF THE NEWTON-RAPHSON METHOD AND THE PROPOSED METHOD USING TRIANGULATION WITH PIVOTING; ON THE IEEE 30 BUS TEST SYSTEM.

	Newton-Raphson	Proposed method
EPS	0.0001	0.0002
Complete iterations	3	5
Iterations just started	1	0
Total number of iterations	4	5
Time for the 1st iteration	1.689 sec.	1.684 sec.
Time for iterations other than the 1st	1.689 sec.	1.476 sec.
Total time to meet EPS	5.127 sec.	7.589 sec.

The same starting values for the voltages have been used with each method. These values are, $e_i^0 = 1.0$ and $f_i^0 = -0.3$. The slack bus has the values $e_s = 1.06$ and $f_s = 0.0$.

The counting of the time includes only the time spent in the iterations. The writing of the scheduled powers and the initial voltages, the calculation of the line flows, and the calculations of the reactive powers at the voltage controlled buses at the end of the problem, are not included.

The proposed method using triangulation with pivoting, converges, but not as fast nor precisely as the Newton-Raphson method. Comments on the precision can be found in section 4.4.

4.2 THE EFFECT OF ACCELERATION FACTORS

Acceleration factors have been tried with the proposed method using triangulation with pivoting for solving the linearized system. The explanation of the convergence acceleration of a load-flow method by means of acceleration factors, can be found in section 8.4 of (3).

The way in which they have been used is by multiplying the bus voltage changes at each iteration by a factor. A pair of different factors were used, one for the real component of the voltage, and another for the reactive. The value of the factors were near unity.

Acceleration factors were tried in two ways: since the first iteration, and from the second iteration on.

The improvement observed with the best pair of acceleration factors, was not enough to get to the solution with less iterations than without acceleration factors.

4.3 COMBINATION OF THE NEWTON-RAPHSON METHOD AND THE PROPOSED METHOD USING TRIANGULATION WITH PIVOTING

It can be seen in appendix 3 that it is very easy to switch from the Newton-Raphson method to the proposed method and back. Taking now advantage of this fact, the 30 bus load-flow problem can be solved by an alternation of the Newton-Raphson method and the proposed method. The method is convergent, faster than the proposed method but not as fast as the plain Newton-Raphson method. The results are:

EPS = 0.00015

number of iterations = 4 (4th iteration complete)

time per iteration: for iterations 1 and 3, (Newton-Raphson) = 1.67 sec.

for iterations 2 and 4, (proposed method) = 1.57 sec.

total time to meet EPS = 6.485 seconds

Other combinations of the Newton-Raphson method and the proposed method were tried. In one of them, a switch to the Newton-Raphson method is done when the value of the changes in voltage at the iterations are less than a specified value. The load-flow problem has been solved with 0.03 as the switch value for the changes of voltage. The results obtained have been:

EPS = 0.0001

number of iterations = 4 (4th iteration complete)

times per iterations: for iters. 1, 3 and 4, (Newton-Raphson) = 1.64 sec.

for iteration 2, (proposed method) = 1.48 seconds

total time to meet EPS = 6.404 seconds

It must be pointed out that the same precision as for the Newton-Raphson has been used.

The load-flow problem has finally been solved implementing the proposed method using triangulation with pivoting but using the Newton-Raphson method when any of the voltage changes at the former iteration is greater than 0.1 . The results are:

EPS = 0.0001

number of iterations = 4 (4th iteration complete)

time per iteration: for iterations 1 and 2 (Newton-Raphson) = 1.63 seconds

for iterations 3 and 4 (proposed method)= 1.52 seconds

total time to meet EPS = 6.132 seconds

4.4 THE WEAK POINTS IN THE PROPOSED METHOD

By analyzing the weak points of the proposed method using triangulation, and those of Newton-Raphson's in section 4.5, the changes to apply to the proposed method can be inferred.

- The rate of convergence decreases as the solution is approached. This means that unless the recalculation of the terms of the linearized system takes a great amount of time at each iteration, the extra iterations needed will offset the savings in time per iteration.

Table 2.1 shows that, in the Newton-Raphson method, the time for the calculation of the jacobian ranges from 47.6% for the 14 bus case to 58.7% of the iteration time for the 57 bus case. With the 30 bus case, the calculation of the new jacobian takes 54.5% of the iteration time. The reason why in table 4.1 only a 12.3% reduction of time per iteration is observed, is that the results of table 2.1 correspond

to a program which only stores in the computer memory the nonzero elements of Y_{bus} , and therefore has to have logical instructions to work with the non-zero elements of Y_{bus} , which are time consuming. The terms of the jacobian are a function of G_{pq} and B_{pq} . ($Y_{pq} = G_{pq} - jB_{pq}$). On the other hand the results in table 4.1 come from a program in which the whole matrix Y_{bus} is stored, so no logical instructions are needed. In section 8.2 the reasons for storing only the nonzero elements of Y_{bus} are discussed.

- Another major inconvenience of the proposed method using triangulation with pivoting is the extra storage required for keeping the coefficients of the linearized system. The solution of a system of linear equations by triangulation with pivoting destroys the matrix of coefficients and the vector of constants. Therefore, for not having to recalculate the whole set of coefficients at each iteration, double storage must be provided. The coefficients of the linearized system mean $[2(NB - S)] \times [2(NB - S)]$ words, where NB is the number of buses and S is the number of slack buses.

The load-flow problem with the proposed method using triangulation with pivoting, and storing only the non-zero elements of Y_{bus} has not been tried because even though the proposed method could be as good or even better than Newton-Raphson method as far as time is concerned, the extra storage for the coefficients of the linearized system offsets any time advantage.

- Another of the inconveniences of the proposed method is that an iteration can be reached where the voltage changes are less than the precision EPS but the bus powers are not yet within EPS

neighbourhood of the scheduled powers.

Successive iterations of the method lead the powers towards the scheduled powers. This would mean a waste of time, because a very little change in voltage would take some whole iterations and there is a risk of getting entangled due to the roundoff error.

A measurement of the mismatch of power can be found in table 6.10, where the mismatch is compared with the changes in voltage in the iterations, and an explanation is given.

4.5 THE WEAK POINTS IN THE NEWTON-RAPHSON METHOD

The disadvantages that will be described below are also disadvantages of the proposed method during the first iteration, due to the fact that the first iteration of the proposed method is as Newton-Raphson's.

The rate of convergence of the Newton-Raphson method is quadratic. But it is not worth spending so much time on the first iteration for solving a linearized system, because beforehand, we know that it is going to yield only an approximate answer.

The time spent on the solution of the linear system ranges from 47.3% of the iteration time in the IEEE 14 bus test system, to 41.2% in the 57 bus test system.

The Newton-Raphson method has to recalculate the elements of the jacobian at each iteration, which takes a long time.

4.6 POSSIBLE IMPROVEMENTS

The results of the thesis show that improvements are indeed possible. The characteristics that an alternative method should have

to give an improvement in time when used with the proposed method are explained below.

4.6.1 THE EFFICIENCY OF THE STEPS OF A METHOD WITH RESPECT TO TIME.

Examining the proposed method it could be seen that if a non-destructive system of solution of a linear system were used, advantage could be taken of not having to recalculate the non-diagonal coefficients of the linearized system at each iteration. Besides a double storage for the coefficients would not have to be provided.

The triangulation with pivoting method for the solution of a system of linear equations is an exact method of solution. It is impossible to obtain with this method an approximate solution using less time, because once the method is started, it must be carried out until the end. Since at the first iteration some of the coefficients are not close to the solution, an exact solution of the linearized equation is a waste of time. An approximate less time consuming solution would be more suitable.

4.6.2 ADVANTAGE OF THE PROPOSED METHOD OVER THE NEWTON-RAPHSON METHOD.

If an approximate method of solution of the linearized system is implemented during the early stages of the load-flow, more iterations will have to be used before arriving to the solution.

If the Newton-Raphson method were employed along with an approximate solution of the linearized system during the early stages, no time improvement would be obtained because, at each iteration, it would be necessary to recompute all the terms of the

jacobian, which alone takes the 58.7% of the iteration time in the Newton-Raphson solution, using triangulation with pivoting, when applied to the IEEE 57 bus test system. Using the proposed method this inconvenience would not exist because, at each iteration, only the diagonal coefficients would have to be recalculated. This time saving due to calculation of the diagonal elements only can be expected to increase with the size of the system.

A time improvement can thus be expected with the proposed method using an approximate technique for the solution of the linearized system.

V. THE THEORETICAL BASIS FOR THE USE OF THE
GAUSS-SEIDEL ITERATIVE TECHNIQUE FOR
SOLVING THE LINEARIZED SYSTEM
IN THE PROPOSED METHOD

5.1 FEASIBILITY OF THE SOLUTION

In this chapter it is shown how the Gauss-Seidel iterative technique for solving a system of linear equations complies with the requirements pointed out in section 4.6, and how the proposed method coupled with the Gauss-Seidel technique can yield results comparable with those obtained from the Newton-Raphson method using triangulation with pivoting.

Sections 7.1 and 7.2 of chapter 7, illustrate how acceleration factors successfully increase the speed of the Gauss-Seidel technique.

5.1.1 REQUIREMENTS FOR CONVERGENCE. A number of sufficient conditions can be found in the literature of numerical analysis, for a linear system to have a convergent iterative solution when using the Gauss-Seidel technique. An explanation of Gauss-Seidel or successive iteration technique can be found in section 4.2 of (11), and some tests of convergence are also presented in section 4.4 of (11).

The convergence conditions are generally based on the fulfilment of some numerical conditions by the elements of the matrix of coefficients. Unfortunately a general theoretical

proof of convergence would be very long and cumbersome, and a numerical proof for a specified system, is useless, because it does not guarantee the convergence for any other case.

In section 5.9 of Conte's book on numerical methods (12), there is a sufficient but not necessary convergence condition for the Gauss-Seidel technique. The use of this condition does not yield a general convergence proof, but applying the particular condition to the proposed method, provides us with some insight into the general convergence properties of the linearized system.

Let us denote l_{pq} as a general term for the matrix of linearized coefficients, which will be called L henceforth. Then a sufficient, but not necessary convergence condition for the Gauss-Seidel technique is that, for any row "p",

$$\sum_q |l_{pq}| < |l_{pp}|, \quad q \neq p \quad (5.1)$$

5.1.2 CHARACTERISTICS OF THE MATRIX OF COEFFICIENTS OF THE LINEARIZED SYSTEM. Equation (3.14d) can be written in the simplified way,

$$\begin{bmatrix} \frac{\Delta P}{\Delta Q} \end{bmatrix} = \begin{bmatrix} \frac{L^I}{L^{III}} & \frac{L^{II}}{L^{IV}} \end{bmatrix} \times \begin{bmatrix} \frac{\Delta e}{\Delta f} \end{bmatrix} ; \text{ where } \begin{bmatrix} \frac{L^I}{L^{III}} & \frac{L^{II}}{L^{IV}} \end{bmatrix} = L \quad (5.2a)$$

The diagonal elements of each submatrix L^I , L^{II} , L^{III} , and L^{IV} , are much bigger than the other elements in the same row; most of the non-diagonal elements are zero. The submatrices are of equal size.

The matrix of coefficients L is of the form (5.2b)

$$\begin{array}{ccccccccc}
 A & a_2 & a_3 & a_4 & a_5 & B & b_2 & b_3 & b_4 & b_5 \\
 c_1 & C & c_3 & c_4 & c_5 & d_1 & D & d_3 & d_4 & d_5 \\
 e_1 & e_2 & E & e_4 & e_5 & f_1 & f_2 & F & f_4 & f_5 \\
 g_1 & g_2 & g_3 & G & g_5 & h_1 & h_2 & h_3 & H & h_5 \\
 i_1 & i_2 & i_3 & i_4 & I & j_1 & j_2 & j_3 & j_4 & J \\
 & & & & & & & & & \\
 K & k_2 & k_3 & k_4 & k_5 & L & l_2 & l_3 & l_4 & l_5 \\
 m_1 & M & m_3 & m_4 & m_5 & o_1 & O & o_3 & o_4 & o_5 \\
 p_1 & p_2 & P & p_4 & p_5 & q_1 & q_2 & Q & q_4 & q_5 \\
 r_1 & r_2 & r_3 & R & r_5 & s_1 & s_2 & s_3 & S & s_5 \\
 t_1 & t_2 & t_3 & t_4 & T & u_1 & u_2 & u_3 & u_4 & U
 \end{array} \tag{5.2b}$$

The diagonal and non-diagonal elements of each part can be calculated as follows,

$$1^I_{pq} = e^o_p G_{pq} - f^o_p B_{pq} \tag{5.3}$$

$$1^I_{pp} = e_p G_{pp} - f_p B_{pp} + C I R_p \tag{5.4}$$

$$1^{II}_{pq} = e^o_p B_{pp} + f^o_p G_{pq} \tag{5.5}$$

$$1^{II}_{pp} = e_p B_{pp} + f_p G_{pp} + C II_p \tag{5.6}$$

$$1^{III}_{pq} = e^o_p B_{pq} + f^o_p G_{pq} \tag{5.7}$$

$$1^{III}_{pp} = e_p B_{pp} + f_p G_{pp} - C II_p \tag{5.8}$$

$$1^{IV}_{pq} = f^o_p B_{pq} - e^o_p G_{pq} \tag{5.9}$$

$$1^{IV}_{pp} = f_p B_{pp} - e_p G_{pp} + CIR_p \quad (5.10)$$

There are some equalities that help in deciding whether the condition (5.1) is satisfied or not.

From the computation technique of the bus admittance matrix, (see Appendix 2), it is evident that

$$G_{pp} = - \sum_q G_{pq} \quad , \quad q = 1, 2, \dots, N \quad (5.11)$$

$$B_{pp} = - \sum_q B_{pq} + \sum_q \frac{b'_{pq}}{2} + c_p \quad , \quad q = 1, 2, \dots, N \quad (5.12)$$

In (5.12) b'_{pq} is the line charging to ground, and c_p would be a synchronous condenser connected to bus "p". These relations can be altered by the presence of a variable tap setting transformer between node "p", and any other node.

From the nature of the expressions (5.3) to (5.10), the range of values and signs that CIR_p and CII_p may have, and the fact that e^0_i , and f^0_i are supposed to be close to the exact values e^*_i , and f^*_i , it can be concluded that, as far as each submatrix is concerned, the diagonal elements will have an absolute value close to the absolute value of the sum of the other terms in the same row.

In case that a bus "p" is connected to a slack bus, the diagonal term will be predominant, i.e.: its absolute value will be bigger than the absolute value of the sum of the terms in the same row. This can be explained by the fact that the equations (5.11) and (5.12) are always satisfied, the terms

that would be called l_{ps} , ("s" standing for slack bus), are missing because they are zero.

Hence, it can be seen that each submatrix L^I , L^{II} , L^{III} , L^{IV} , complies with the condition (5.1), and for the rows that correspond to a bus directly connected to a slack bus, the condition (5.1) is clearly satisfied. However, the problem deals with L as a whole, therefore we shall see how L can satisfy the condition (5.1) as well.

Each one of the submatrices L^I , L^{II} , L^{III} , and L^{IV} , has predominant diagonal elements. For the application of the ordinary Gauss-Seidel technique the convergence is guaranteed if the main diagonal elements of L are dominant. For a bus "p" it could be that l_{pp}^I and l_{pp}^{IV} , which have similar value, are more predominant than l_{pp}^{II} and l_{pp}^{III} , whose numerical values are close to each other. Although condition (5.1) is not necessary for the convergence of the Gauss-Seidel technique, applying a procedure that is capable of transforming the linearized system in such a way that condition (5.1) is satisfied, will bring about better numerical results.

5.1.2.1 The elimination of the diagonal of L^{III} . The satisfaction of the condition (5.1) could be improved in several ways if the diagonal terms of L^{II} and L^{III} were to be more dominant than those of L^I and L^{IV} . One of the ways would be by the rotation of the submatrices, putting L^{II} at the top left side and L^I at the top right side, and then L^{IV} at the

bottom left side and L^{III} at the bottom right side. The terms of the vector of increment of voltage, that come from the solution of the linearized systems, would then be in a different order: $\Delta f_1, \Delta f_2, \dots, \Delta f_N, \Delta e_1, \Delta e_2, \dots, \Delta e_N$. This method has not been tried because other less complicated methods, as far as computer programming is concerned, have proved to work well enough.

The Gauss-Seidel method is a successive displacement method, which means that the recently calculated elements of the solution vector are successively used as they are obtained.

Since the diagonal elements are predominant in all four submatrices, it is our purpose to find an algorithm such that a simultaneous solution for Δe_i and Δf_i could be found with it. In this way the error that would be brought about by multiplying the big diagonal elements by the inexact assumed values coming from the former run, would be lessened.

The algorithm used for the implementation of the Gauss-Seidel technique, and for approaching the linearized system to the satisfaction of the condition (5.1), consists of the successive elimination of the diagonal elements of L^{III} . The elimination is nothing but a set of equivalent transformations of the original linearized system.

For illustration, the operations will be made on the matrix pattern (5.2b). At first, the first diagonal element of L^{III} , K , will be eliminated by subtracting from its row, the first row of

the matrix L , where the diagonal term A lies, multiplied by the factor K/A . Then the increment Δf_1 is calculated by multiplying all the elements of the transformed row except $(L - KxB/L)$, which is the transform of element L , by the estimations Δe_i , and Δf_i , on a ordered, element by element basis. Then the sum of all these products is subtracted from the transformed ΔQ_1 . Dividing the subtraction by $(L - KxB/L)$, a new value for Δf_1 is obtained. So far no diagonal element has been multiplied by any Δe_i or Δf_i , because K was eliminated.

Δe_1 is calculated by multiplying $a_2, a_3, a_4, a_5, B, b_2, b_3, b_4, b_5$, by $\Delta e_2, \Delta e_3, \Delta e_4, \Delta e_5, \Delta f_1, \Delta f_2, \Delta f_3, \Delta f_4, \Delta f_5$, respectively. The products are then added up, and the addition subtracted from ΔP_1 . The result is then divided by A thus obtaining the new value Δe_1 . It can be noticed that in the calculation of Δe_1 , one big diagonal element, B , has been multiplied, but it has been multiplied by Δf_1 , which was calculated just at the step before, therefore being closer to the solution than the previous value.

The same is now done with the second diagonal element of L^{III} . M is eliminated by subtracting the second row of the matrix multiplied by M/C ; Δf_2 is calculated, and then a new Δe_2 is computed using already the values of Δf_2 , and Δe_1 and Δf_1 , obtained in the previous step. The same process is repeated until a whole new set $\Delta e_i, \Delta f_i$ has been calculated, and so completed a run of the Gauss-Seidel technique.

Since the constant elements $\Delta Q_1, \Delta Q_2, \dots, \Delta Q_N$, are modified

during the elimination of the elements of the diagonal of L^{III} , and they are long to compute, an extra vector to keep those values must be supplied.

5.1.2.2 Characteristics of the method. The Gauss-Seidel technique, using the algorithm described above, has some advantages.

- The algorithm is very easy to program, because it is only a sequence of operations, which can be solved in a single loop.

- In case that the diagonal terms l_{pp}^{II} and l_{pp}^{III} are more dominant than l_{pp}^I and l_{pp}^{IV} , the elimination of the diagonal of L^{III} yields an equivalent system. In this system the new l_{pp}^{IV} elements are more predominant than before, because the new l_{pp}^{IV} is a linear combination of the ancient l_{pp}^{IV} and l_{pp}^{II} , which is more diagonal dominant. It can be said that, since the rows of the lower half of L are transformed into a row with only one dominant element, the condition (5.1) will be satisfied more largely.

- The more slack buses are used, the faster the convergence of the Gauss-Seidel technique. As explained before, the buses connected to a slack bus have the diagonal term predominant, because in its row, the term corresponding to the slack bus is missing. This is an advantage over the triangulation with pivoting technique, where the number of slack buses does not affect the speed of solution.

- Another advantage over the triangulation with pivoting

technique is that, as stated in section 4 of chapter 2 in reference (11), errors due to roundoff and blunders may be damped out as the Gauss-Seidel technique proceeds.

5.2 THE APPLICATION OF AN ITERATIVE TECHNIQUE FOR SOLVING THE LINEARIZED SYSTEM.

Experimental results with the IEEE 14, 30, and 57 bus test systems show that the Gauss-Seidel technique is convergent but its speed is very low. Acceleration factors have proved to increase the speed of the convergence of the Gauss-Seidel technique for solving the linearized system. Even though the speed of convergence is low, the proposed method with the Gauss-Seidel technique gives faster results than the Newton-Raphson method using triangulation with pivoting, in some cases.

5.2.1 DIFFERENCE BETWEEN ITERATION AND RUN. In order to avoid confusion a different name has been used for the iterations in the proposed method or in the Newton-Raphson method, (which have been called "iterations" all along), and for the iterations in the Gauss-Seidel technique for solving the linearized system. Each of the iterations, namely, the calculation of a new set of values Δe_i , Δf_i , has been called a "run". For instance, we could use 10 "runs" of the Gauss-Seidel technique at each iteration, which would mean that the process of obtaining a new set of values Δe_i , Δf_i , would be applied ten times between two changes of the diagonal elements of the coefficient matrix.

The number of "runs" can be constant, or variable, or regulated by a criteria, which measures the rate of convergence of the results in the runs, and stops the calculations in a new run when the rates of convergence reach a specified value.

When initiating an "iteration", a check of the convergence in voltage and in power is made, and the values of the voltage and the currents in the diagonals of the linearized system matrix L are updated. Once this is done, the program goes to the routine for solving the new linear system. If this routine is a Gauss-Seidel technique such as described above, a certain number of "runs" will be used to get a new, better approximation of the solution.

When the Gauss-Seidel technique is used for solving the system of linear equations, a starting set of values must be supplied. In the proposed method the set of starting values is only supplied at the first iteration. These values are $\Delta e_i = 0.0$, $\Delta f_i = 0.0$, because it is not known in which direction, with respect to the initial values e_i^0 , f_i^0 , the solution is going to be. During the iterations other than the first, the result of the former iteration is the best starting value that can be used.

If the Gauss-Seidel technique along with the Newton-Raphson method had been used, the starting values would have to be zero at the beginning of each iteration, because the increments at the beginning of a new iteration have nothing to do with the increments of voltage in the former iteration.

obtained. Applying this process a few times, the precision obtained in the first iteration using either triangulation with pivoting or a high number of runs per iteration with the Gauss-Seidel technique, is reached, and the time elapsed proves to be less.

Another advantage is the possible application of acceleration factors, which can be implemented at any stage of the Gauss-Seidel technique. In chapter 7 is shown that the acceleration factors speed up the solution.

5.2.2 TENDENCY OF THE RESULTS OBTAINED IN THE RUNS. As it was pointed out before, it is a waste of time to solve the linearized system with high precision at the first iteration because the values CIR_p and CII_p are far from the final values. Time would be spent for solving with accuracy an approximate system. Even being solved exactly, it would only give the tendency towards the solution.

It has been found that the same tendency towards the solution can be obtained in less time using the Gauss-Seidel technique for solving the linearized system, but using only a few runs instead of a complete solution, which would take a lot of runs. The complete solution by the Gauss-Seidel technique takes a much longer time than the triangulation with pivoting technique, due to the low speed of convergence.

Advantage can be taken of two facts: first, whereas the triangulation technique has to be carried out completely no matter how much accuracy is needed, with the Gauss-Seidel technique, the calculation of "runs" can be stopped at any time, so, if no high precision is needed, a few runs will be enough. Of course the voltages are not as close to the solution after a small number of runs as they would have been if more runs per iteration or a triangulation routine had been used. Since the tendency of the results of the runs is always towards the solution, as the system is convergent, if the currents CIR_p and CII_p are recomputed, better values for them would be obtained. Substituting them in the diagonal elements of the linearized system, a system that will yield an improved solution is

VI. RESULTS OF THE PROPOSED METHOD USING THE GAUSS-
SEIDEL ITERATIVE TECHNIQUE FOR SOLVING THE
LINEARIZED SYSTEM

6.1 ANALYSIS OF THE RESULTS OF AN EXACT, AND AN APPROXIMATE
SOLUTION OF THE LINEARIZED SYSTEM IN THE PROPOSED METHOD

In section 5.2.2 it was reported that an approximate iterative solution of the linearized system would yield time advantages. The approximate solution is obtained by using only a few runs of the Gauss-Seidel technique instead of as many as necessary for the exact solution of the linearized system at each iteration.

For the solution of the linearized system at each iteration a different number of runs could be used, but only the results with a constant number of runs per iteration will be shown. One of the improvements of the proposed method could be the increasing of the number of runs per iteration at each iteration, so as to obtain more accurate results as the solution is approached. However, even a constant number of runs per iteration improves the solution time.

All the results use the same degree of precision so that a fair comparison can be drawn. It must be pointed out that when using the IBM library subroutines CS019A and CS019B for measuring the time spent, account has to be taken of several facts: first, the amount of output. All printing operations are time consuming, and the input-output modes use a variable time depending on the circumstances. To avoid that each case has been solved

twice, one time with printing of the intermediate results, and another without any printing until the final results are reached. Even so, sometimes the time employed for the same problem is different due to the fact that the computer works on a shared time basis. To overcome that there has been run along with all the cases a reference case, which consists of the Newton-Raphson solution.

The proposed method for solving the load-flow problem is convergent for the IEEE 14, 30 and 57 bus test systems, and, besides, as specified in section 5.12, the iterative technique for the solution of the linearized system is convergent too. Several cases using different runs per iteration are shown for each test system.

Since the results of the load-flow problem for each system is known, the convergence at each step of the solution can be checked by finding the differences between the final voltages and the voltages at each stage of the solution.

6.1.1 RESULTS FOR THE 14 BUS TEST SYSTEM.

In table 6.1 is shown that there exists a certain number of runs per iteration which gives the solution in minimum time with respect to other numbers of runs per iteration. This is so because as the number of runs per iteration decreases, the time per iteration decreases, but the number of iterations for convergence increases. The minimum is the best compromise.

TABLE 6.1 TIMES AND NUMBER OF ITERATIONS FOR SEVERAL CASES OF APPROXIMATE SOLUTION OF THE LINEARIZED SYSTEM IN THE PROPOSED METHOD APPLIED TO THE 14 BUS SYSTEM.

runs per iteration	iterations for convergences	time for solution (seconds)
150	6	10.7
40	6	4.54
20	6	2.237
13	5	1.488
10	7	1.641
8	10	1.918
6	12	1.801
5	14	1.811
4	17	1.97
3	22	2.144
2	30	2.132
1	50	2.3
Newton-Raphson	4	0.624

N.B. The program used deals with the whole Y_{bus} matrix, i.e.: there are no logical instructions in the program to deal with only the non-zero elements of Y_{bus} .

Some of the maximum and minimum active and reactive voltage error with respect to the final values are shown in table 6.2.

It can be observed that the error at the end of each iteration gets bigger as the number of runs per iteration used decreases.

TABLE 6.2 MAXIMUM AND MINIMUM VOLTAGE ERRORS AT EACH ITERATION OF THE PROPOSED METHOD WITH SEVERAL

CASES OF APPROXIMATE SOLUTIONS OF THE LINEARIZED SYSTEM. APPLICATION TO THE 14 BUS TEST

SYSTEM.

N.B. Each case at each iteration is represented by a group of four numbers. At the top left is found the maximum difference in real voltage, at the bottom left its minimum; the maximum reactive voltage difference is at the top right position, and its minimum is at the bottom right. All the quantities are per unit voltages multiplied by 10^5 .

runs per iteration	ite. 1	ite. 2	ite. 3	ite. 4	ite. 5	ite. 6
N-R	-623	-1022	-23	-18	0	0
	-2678	-2204	-47	-74	0	0
10	367	2035	72	459		
	-2324	-360	-68	-375		
8	510	2580	192	946	114	453
	-2250	21	-66	-218	49	130
6	690	3223	364	1628	245	957
	-2125	651	-70	0	53	274
4	937	4046	630	2583	453	1799
	-1845	1752	-101	358	60	508
					355	1323
					23	407
2	1207	6179	1054	4049	711	2766
	-1214	2791	-65	1624	48	859
					631	2344
					551	2004
					46	715
					41	610

The results in table 6.1 reveal that even with only one run per iteration the proposed method is convergent. The convergence speed decreases as predicted according to the theoretical results in sections 3.1.1 and 3.3.

The time spent using 10 and 4 runs per iteration is shown in table 6.3. The program used deals only with the nonzero elements of the matrix Y_{bus} , therefore the program has logical instructions to work only with those elements. The results could be compared with those in table 2.1, which correspond to the Newton-Raphson method.

TABLE 6.3 TIME SPENT BY THE PROPOSED METHOD USING 10 AND 4 RUNS PER ITERATION WHEN APPLIED TO THE 14 BUS TEST SYSTEM

10 runs per iteration, times in seconds

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	0.3927	0.0175	0.1613	0.2099
2-7	0.2312	0.0169	0.0028	0.2115

4 runs per iteration, times in seconds

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	0.2656	0.0177	0.1613	0.0866
2-17	0.1074	0.0171	0.0041	0.0863

N.B. The times spent for the calculation of the new currents and powers, and for the solution of the linearized system and

calculation of the new voltages ought to be the same at each iteration for each case; however, small variations are observed. Moreover, small variations are observed as well in the time spent on the different items of the iterations beyond the first one.

Comparing the times taken at each step of the iteration, as shown in table 6.3, with the times taken by the Newton-Raphson method, the following can be concluded:

- The time for computing the new currents and powers is more or less the same using the Newton-Raphson method and the proposed method, and in the last, it is independent of the number of runs per iteration, as expected.

- In the proposed method, the time for the computing of the new elements of the linearized system is long in the first iteration and short in the others. The fact of calculating only the diagonal elements after the first iteration accounts for it. The time spent on this step in the first iteration of the proposed method is more or less the same as the corresponding time for the Newton-Raphson method. Here lies one of the advantages of the proposed method: this method does not spend time in recalculating all the coefficients of the linearized system at each iteration.

- The time for solving the linear system is variable depending on the number of runs per iteration: 0.2099 s. for 10 runs per iteration, and 0.0866 s. for 4 runs per iteration. The triangulation with pivoting used with the Newton-Raphson method takes 0.1662 s. . Therefore, depending on the number of runs per

iteration, time will be saved or lost with respect to the Newton-Raphson method.

- A comment on the precision can be found with the 57 bus case.

6.1.2 RESULTS FOR THE 30 BUS TEST SYSTEM

TABLE 6.4 TIMES AND NUMBER OF ITERATIONS FOR SEVERAL CASES OF APPROXIMATE SOLUTIONS OF THE LINEARIZED SYSTEM IN THE PROPOSED METHOD APPLIED TO THE 30 BUS SYSTEM.

runs per iteration	iterations for converg.	time for solution (seconds)
150	6	-
40	7	-
20	9	-
13	11	14.09
10	15	15.03
8	18	15.08
6	23	14.70
5	26	14.49
4	31	14.16
3	38	14.08
2	49	13.64
1	73	13.83
N-R	4	4.45

The same N.B. as in Table 6.1 applies here.

Table 6.5 shows the voltage errors for this test system.

The times obtained when using the proposed method with 5 runs per iteration are shown in table 6.6 .

TABLE 6.5 MAXIMUM AND MINIMUM VOLTAGE ERRORS AT EACH ITERATION OF THE PROPOSED METHOD WITH SEVERAL
CASES OF APPROXIMATE SOLUTIONS OF THE LINEARIZED SYSTEM. APPLICATION TO THE 30 BUS TEST
SYSTEM.

The same N.B. as in table 6.2 applies here.

runs per iteration	ite. 1	ite. 2	ite. 3	ite. 4	ite. 5	ite. 7
N-R	-635 -963	-35 -13	0 0			
	-2484 -2508	-66 -81	0 0			
10	434 2201	240 1266	179 887			
	-1874 233	-268 -95	-100 212			
8	495 2534	319 1563	127 1183	190 899		
	-1774 752	-256 62	-91 293	-21 232		
6	556 3135	400 1911	327 1498	269 1249	223 1035	
	-1633 1450	-268 333	-92 428	-15 335	19 266	
4	712 4571	510 2489	427 1955	373 1623	326 1470	290 1315
	-1428 1114	-426 863	-137 733	-37 541	16 428	29 359
2	1194 7688	879 4334	652 3052	534 2528	488 2257	451 2022
	-1891 490	-1174 1097	-623 1435	-299 1234	-122 972	-57 796

TABLE 6.6 TIME SPENT BY THE PROPOSED METHOD USING 5 RUNS PER
ITERATION WHEN APPLIED TO THE 30 BUS TEST SYSTEM.

Times in seconds

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	2.0566	0.0527	1.5603	0.4436
2-26	0.5028	0.0518	0.0057	0.4453

The same N.B. as in table 6.3 applies here.

Comparing the results in Table 6.6 with the results of the Newton-Raphson method for the same system, the same conclusions as for the 14 bus test system would be reached.

6.1.3 RESULTS FOR THE 57 BUS TEST SYSTEM

TABLE 6.7 TIMES AND NUMBER OF ITERATIONS FOR SEVERAL CASES OF
APPROXIMATE SOLUTIONS OF THE LINEARIZED SYSTEM IN THE
PROPOSED METHOD APPLIED TO THE 57 BUS SYSTEM.

runs per iteration	iterations for converg.	time for solution	time for first ite.	time for iteration (seconds)
24	14	122.8	19.74	8.00
18	18	125.2	17.9	6.16
12	25	121.0	15.85	4.16
10	28	106.0	14.9	3.35
8	34	111.0	14.54	2.9
6	42	103.0	13.83	2.165
5	48	102.9	13.48	1.9
4	56	96.3	13.2	1.5

runs per iteration	iterations for converg.	time for solution	time for first ite.	time for iteration (seconds)
2	-	-	12.2	0.835
N-R	4 (3 compl.)	60.0	19.88	19.88

The program used deals only with the non-zero elements of Y_{bus} by means of logical instructions.

It can be observed in table 6.7, that the times for the first iteration, are much higher than those for the other iterations. The reason is that during the first iteration all the elements of the linearized system must be calculated.

The voltage errors for the 57 bus test system are found in table 6.8 .

TABLE 6.8 MAXIMUM AND MINIMUM VOLTAGE ERRORS AT EACH ITERATION OF THE PROPOSED METHOD WITH SEVERAL CASES OF APPROXIMATE SOLUTIONS OF THE LINEARIZED SYSTEM. APPLICATION TO THE 57 BUS TEST SYSTEM.

The same N.B. as in table 6.2 applies here.

runs per iteration	ite. 1		ite. 2		ite. 3		ite. 4		ite. 5	
N-R	-1301	-803	-63	-29	0	0				
	-4782	-2836	-212	-142	0	0				
10	935	9348	661	7380	460	5664	333	4784		
	-6770	458	-4511	-336	-3280	460	-2647	378		
8	1196	9876	863	8214	618	6181	464	5488	355	4803
	-7357	670	-5243	-173	-3855	584	-3026	488	-2532	353

runs per iteration	ite. 1		ite. 4		ite. 8		ite. 16	
6	1497	10528	662	6527				
	7993	969	-3787	646				
4	1822	11393	989	8104	518	5512		
	-8680	651	-5165	888	-2826	477		
2	2468	12551	1599	9777	1037	8098	541	5532
	-9412	208	-7315	1373	-5120	877	-2727	487

TABLE 6.9 TIME SPENT BY THE PROPOSED METHOD USING 12 RUNS PER
ITERATION WHEN APPLIED TO THE 57 BUS TEST SYSTEM.

Times in seconds

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	15.2760	0.1448	11.2762	3.8550
2-25	4.0192	0.1434	0.0110	3.8648

The same N.B. as in table 6.3 applies here.

6.2 COMPARISON BETWEEN THE RESULTS IN THE THREE SYSTEMS AND CONCLUSIONS

The following facts can be observed:

- The values of the errors at each iteration for the 57 bus case are much higher than for the 14 and 30 bus cases. This is due to several reasons. At first, the initial guess $E_k = 1.0$, $F_k = -0.3$, is not as good in this case as it is for the 14 and 30 bus case. It is very likely that $F_k = -0.22$ would have worked better, for this value is closer to the average final value for the reactive voltage. Moreover the index number of lines/number of nodes is

1.403, (1.366 for the 30 bus system), which affects the error at the end of the first iteration. Indeed, during the first run of the first iteration of the proposed method, a starting value of zero for all the increments is used. Therefore, the more lines connected to each bus, the bigger the relative error is at the end of the first run.

- The rate of convergence of the proposed method is decreasing, and at the end is much slower than that of the Newton-Raphson method. However, since the time spent for iteration is smaller, there is a period of time in which the proposed method gives better results than the Newton-Raphson method. See section 3.1.3.

- A consequence of the decreasing convergence is that much time is spent on the last iterations, but there is not much improvement of the solution. The program cuts off the problem when the changes in voltage are less than the precision used, i.e.: 0.0001. This is done in order not to spend time on an improvement, which is very slow and is likely to be overcome in part by round-off.

- Another consequence of the decreasing convergence is that although the increments of voltage are less than the tolerance 0.0001 when the solution is cut off, the precision in power might not be below the tolerance. The accumulated numerical error in the diagonal terms of the linearized system, due to a high number of iterations, could account for it.

A display of the maximum differences in power and the maximum changes of voltage at the last iterations, can be found in table 6.10.

TABLE 6.10 MAXIMUM DIFFERENCES IN POWER AND BIGGEST VOLTAGE CHANGE AT THE FINAL ITERATIONS
OF THE 12 RUNS/ITE. CASE OF THE 57 BUS TEST SYSTEM AND THE 6 RUNS/ITE. CASE OF THE
14 BUS TEST SYSTEM.

TMX stands for the maximum change in voltage and SMX for the maximum difference in power. 57 bus
test system, 12 runs/ite.

	ite. 20	ite. 21	ite. 22	ite. 23	ite. 24	ite. 25
TMX	0.00026	0.00021	0.00017	0.00013	0.00010	0.00008
SMX	0.00091	0.00066	0.00047	0.00033	0.00023	0.00018

14 bus test system, 6 runs/ite.

	ite. 7	ite. 8	ite. 9	ite. 10	ite. 11	ite. 12
TMX	0.00090	0.00057	0.00036	0.00023	0.00015	0.00009
SMX	0.00380	0.00241	0.00153	0.00097	0.00062	0.00039

From the results in table 6.10 the following can be concluded:

- a) The convergence is indeed slower at the end of the process.
- b) There is a trend towards a bigger mismatch of power as the number of runs per iteration decreases. This phenomenon is general.
- c) Since TMX is measured after the corrections of voltage and SMX is measured before, the mismatch is not as big as it appears because the value of SMX at the iteration "n+1" would in fact correspond to the value of TMX at the iteration "n", and SMX decreases.
- d) As a consequence of a), b), and c), the convenience of finishing off with one iteration using the Newton-Raphson method, can be seen. This iteration would eliminate the power mismatch, and would converge more rapidly.

VII. IMPROVEMENTS IN THE PROPOSED METHOD

7.1 THE USE OF ACCELERATION FACTORS IN THE SOLUTION OF THE LINEARIZED SYSTEM BY THE GAUSS-SEIDEL TECHNIQUE

Acceleration is only implemented in the solution of the linearized system at each iteration, so as to obtain more exact results in less iterations. By doing so, the extra time that would have been taken by the non-accelerated iterations to reach the same precision, is saved.

Other acceleration procedures could be used between the values at the end of each iteration, but these are the ordinary techniques that are used with every direct load-flow method, and they will not be used with either the proposed method nor with the Newton-Raphson method, which is taken as the reference case.

Some values of acceleration factors have been tried. The values used improve the speed more than other values tested, but it is possible they may be improved by another set of values. A search for the best values has not been intended because the speed of the process can still be improved in many other ways, and besides, the optimum values would be different for each system. The goal was to show that acceleration factors improve the solution, sometimes to such an extent that we reach the solution of the load flow problem in less time than the Newton-Raphson method.

7.1.1 ACCELERATION FORMULA EMPLOYED.

At the end of each run the iterative technique comes up with a set of increments of voltage

with respect to the first guess. They will be called DFT_k . The values of DFT_k are stored for three consecutive runs, which will be called: $r-2$, $r-1$, and r . Then a check is made to see if, for every component, DFT_k^{r-1} lies between DFT_k^{r-2} and DFT_k^r . If this is so, the value of the component is modified according to the formula (7.1).

$$DFT_k^{r \text{ new}} = DFT_k^r + (\text{accl. factor}) \times (DFT_k^r - DFT_k^{r-1}) \quad (7.1)$$

A different acceleration factor is used for the components that correspond to the active voltages, and those corresponding to the reactive voltages. Details about acceleration factors can be found in section 2.5 of reference (11).

The acceleration formula (7.1) is one of those possible. Due to the checking that is made to see whether DFT_k^{r-1} lies between the values at the iteration before and after, the components of DFT whose values oscillate, are not modified.

7.2 ANALYSIS OF THE EFFECT OF ACCELERATION FACTORS IN THE SOLUTION OF THE LINEARIZED SYSTEM

The effect of acceleration factors is shown by the performance of the proposed method using several values for them. FRL will be the acceleration factor for the real part of the voltage increments, and FIM that for the imaginary part of the voltage increments. When the value of the acceleration factors is $FRL = 0.0$, and $FIM = 0.0$, it is as if no acceleration factors were applied to the iterative technique of solution of the linearized system.

In table 7.1 several pairs of acceleration factors are tried on the 3 runs/iteration case of the solution of the 57 bus test system by the proposed method. It can be seen that for the same iteration, the pair FRL 3.5 FIM 5.0 gives the lowest error limits.

A graph of the acceleration factor effects is shown in figure 7.1.

For the 30 bus test system, the sensitivity with respect to the acceleration factors, is shown by the number of iterations and time to convergence. In table 7.2, the 6 runs per iteration case is shown. It can be noticed that three of the acceleration factors tried, yield the solution in less time than the Newton-Raphson method.

TABLE 7.1 EFFECT OF ACCELERATION FACTORS ON THE ERROR IN VOLTAGE OF THE 3 RUNS/ITERATION SOLUTION OF

THE BUS 57 BUS TEST SYSTEM WITH THE PROPOSED METHOD.

The same N.B. as in Table 6.2 applies here.

accel. factor	ite. 1	ite. 2	ite. 3	ite. 8	ite. 12	ite. 20						
FRL=0.0	2086	11955	1778	10495	1472	9498	715	6581	462	3199		
FIM=0.0	-9042	423	-7980	667	-7013	1217	-3663	634	-2475	425		
FRL=2.2	1946	13094	1157	10228	793	7101	226	3550	94	1908	10	367
FIM=4.0	-8229	-1462	-6594	20	-5249	870	-2640	209	-1916	90	-899	10
FRL=2.8	2657	13499	1095	10527	795	6518	168	2990	59	1300	-1	146
FIM=5.0	-8007	-2271	-6262	115	-4851	891	-2496	157	-1719	58	-688	-18
FRL=3.5	2265	13499	1117	10481	791	6499	166	3095	63	1391	3	191
FIM=5.0	-7748	-2271	-5817	30	-4287	889	-2145	161	-1465	63	-545	4

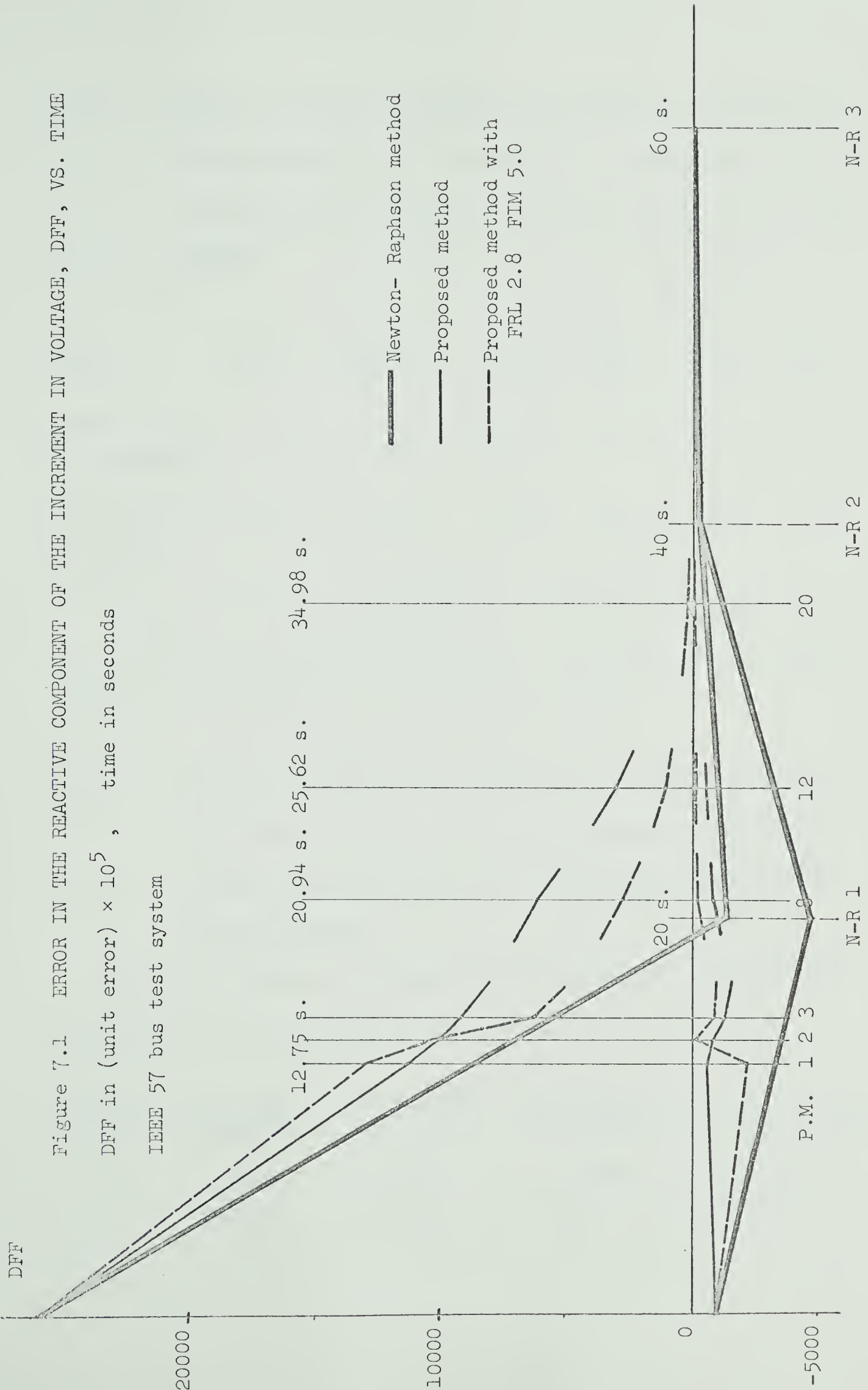


Figure 7.1 ERROR IN THE REACTIVE COMPONENT OF THE INCREMENT IN VOLTAGE, DFF, VS. TIME
DFF in (unit error) $\times 10^5$, time in seconds
IEEE 57 bus test system

TABLE 7.2 EFFECT OF SEVERAL ACCELERATIONS FACTORS ON THE NUMBER OF ITERATIONS AND TIME TO CONVERGENCE. APPLICATION TO THE 6 RUNS/ITERATION CASE OF THE SOLUTION OF THE 30 BUS TEST SYSTEM.

case	N-R	FRL 0.0 FIM 0.0	4.0 8.0	3.0 6.0	3.0 6.5	3.0 5.5
number of ite. for convergen.	4	23	20	11	11	10
time for conv. (in seconds)	8.64	16.51	15.51	8.00	8.00	7.71

In table 7.3 there are the results of several cases of solution of the 14 bus test system by the proposed method using acceleration factors for the solution of the linearized system.

TABLE 7.3 EFFECT OF ACCELERATION FACTORS ON THE TIMES OF SOLUTION AND NUMBER OF ITERATIONS FOR CONVERGENCE. APPLICATION TO THE 4 RUNS/ITERATION CASE OF THE SOLUTION OF THE 14 BUS TEST SYSTEM.

Solution by the Newton-Raphson method using triangulation with pivoting for the linearized system. Total time is 0.9955 s.

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1-3	0.3262	0.0167	0.1551	0.1544
4	0.0167	0.0167	0.0	0.0

Proposed method, 4 runs/iteration. FRL = 3.0 , FIM = 6.0 .

Total time 1.4034 s.

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	system solution and new volts.
1	0.2569	0.0167	0.1561	0.0840
2-12	0.1041	0.0163	0.0031	0.0847

Proposed method, 4 runs/iteration. FRL = 3.0 , FIM = 6.0 .

Total time 0.9868 s.

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	0.2568	0.0167	0.1561	0.0840
2-8	0.1041	0.0163	0.0031	0.0847

Proposed method, 4 runs/iteration. FRL = 4.0 , FIM = 8.0 .

Total time 1.4029 s.

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	0.2559	0.0167	0.1551	0.0840
2-12	0.1041	0.0163	0.0031	0.0847

The same N.B. as in Table 6.3 applies here.

7.3 COMMENTS ON THE USE OF ACCELERATION FACTORS

As can be supposed from the acceleration formula, the subroutine which performs the acceleration works as soon as three successive sets of DFT are given to the subroutine. One set is given to the sub-

routine each time that it is called. The counts are reset to zero whenever a new iteration is started. Thus, if 7 runs per iteration are made, the acceleration formula works twice; at the end of the third run, and at the end of the sixth run. The subroutine stores the first set of DFT at the seventh run. But as at the next iteration, the counting of the sets of DFT stored is reset at 0, the first set of DFT, which now is that obtained at the first run of the new iteration, is stored again.

With each test system (14, 30, 57 buses), there is a number of runs per iteration such that the total time for convergence is less than the time spent by Newton-Raphson method, provided that a certain pair of acceleration factors. As an example the following values are suggested.

- for 57 bus test system: 6 runs/ite. along with FRL 3.5 and FIM 5.0
- for 30 bus test system: 6 runs/ite. along with FRL 3.0 and FIM 5.5
- for 14 bus test system: 4 runs/ite. along with FRL 3.0 and FIM 6.0

The results however are liable to have a small mismatch in power, due, as when no acceleration factors were used, to the cutting off when the voltage increments are less than the precision (0.0001).

Although the solution is reached faster if the suitable number of runs per iteration and acceleration factors are used, the solution still is not good enough, because of the slow power convergence

during the final iterations, and the final power mismatch. In the next section it will be shown how a final iteration using the Newton-Raphson method reduces the time for solution and eliminates the power mismatch.

7.4 ITERATIVE APPROXIMATE SOLUTION AND SWITCH TO AN EXACT SOLUTION AND TO THE NEWTON-RAPHSON METHOD

There is no advantage in spending much time on the solution of the linearized system during the early iterations because it is known beforehand that the solution is only an approximation at that stage. The approximate solution of the linearized system in the proposed method tries to postpone an exact, time-consuming solution of the linearized system until the voltages are close to the final results.

When the results get close enough to the solution, an exact method of solution can be applied instead of the approximate iterative technique. The iteration in which the change of method of solution of the linearized system takes place, will be called ISW.

A procedure that greatly improves the performance of the proposed method is the following: when the iteration ISW is reached, the linearized system is solved by triangulation with pivoting instead of solving it by the Gauss-Seidel iterative technique. From the next iteration on, the Newton-Raphson method is used. The following principles have to be borne in mind.

- During the first iteration and the early ones, only a few runs per iteration will be used. This will give a new set of voltages

not as close to the solution as would have been obtained using an exact solution. However, since the time spent is very short, the process can be repeated several times, and come up, at the end, with a better approximation than that obtained with an exact time consuming solution. The time spent for this is less than that for the exact solution.

- The process explained before is repeated until the values of the voltages are such that, when applying an exact solution, it gives a set of values of the voltages which go to the final solution with a single iteration of the Newton-Raphson method. A previous run of the program without any switching to the Newton-Raphson method could be made to see which is the more convenient iteration to make the switch to the exact solution. However it is also possible to let the problem itself find out when to switch to the exact solution. The maximum difference of power at the beginning of the iteration, or the maximum voltage change at the end of it are a good indication of whether the voltages are close or not to the final solution.

- The exact method used for solving the linearized system at the iteration ISW is triangulation with pivoting, which means destruction of the matrix of coefficients. In this case it does not matter because it had to be recalculated with the latest values of the voltages so that the Newton-Raphson technique can be used. The matrix of coefficients dealt with was that calculated at the first iteration with the assumed voltages, except for the diagonal terms, which were updated at each iteration.

- This method of switching to the Newton-Raphson technique is one way to improve the solution, but there are several others. One, that could be tried, is increasing at each iteration the number of runs per iteration. Both the power mismatch and the slow convergence would likely be reduced without a great consumption of time.

The use of switching is independent of acceleration techniques for solving the linearized system.

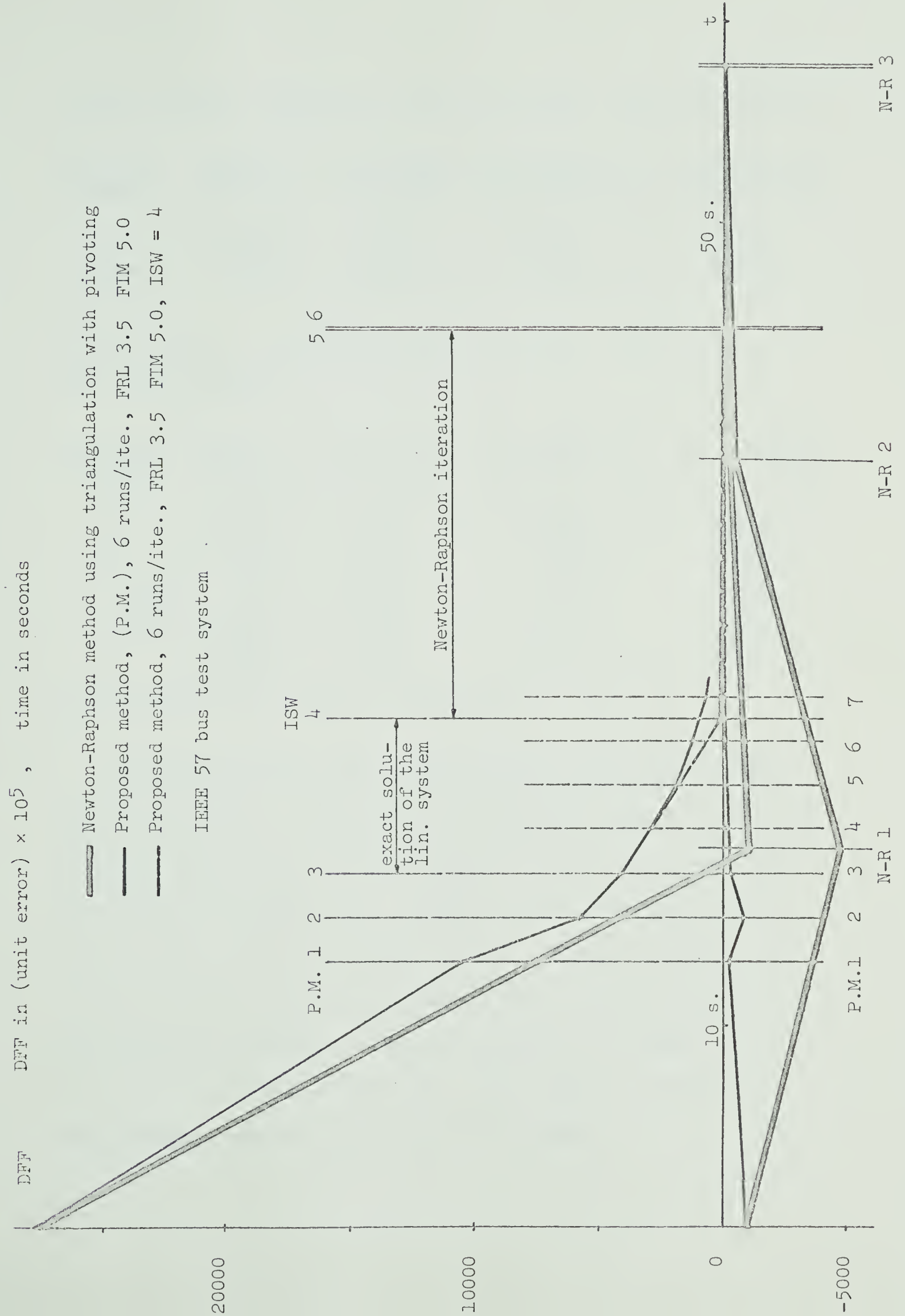
In figure 7.2 is displayed; the performance of the Newton-Raphson method, of the proposed method with 6 runs/ite. and FRL 3.5 and FIM 5.0 , and of the same method with switch at the iteration ISW = 4, on the 57 bus test system. In table 7.4 are shown the times and iterations spent by each method.

TABLE 7.4 NEWTON-RAPHSON METHOD, PROPOSED METHOD WITH 6 RUNS/ITE.
AND FRL 3.5 AND FIM 5.0, AND THE SAME WITH SWITCH AT
ISW = 4. TIMES AND ITERATIONS FOR SOLUTION OF THE 57
BUS TEST SYSTEM

Solution by the Newton-Raphson method using triangulation with pivoting for the linearized system. Total time 57.8403 s.

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1-3	19.2078	0.1592	11.2905	7.7670
4	0.1492	0.1492	0.0	0.0

Figure 7.2 ERROR IN THE REACTIVE COMPONENT OF THE INCREMENT IN VOLTAGE, DFF, VS. TIME



Proposed method, 4 runs/ite. FRL 3.5 FIM 5.0 . Total time 56.8966 s.

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	13.7952	0.1579	11.6366	2.0006
2-21	2.1493	0.1488	0.0113	1.9892

Proposed method, 4 runs/ite. FRL 3.5 FIM 5.0 . ISW = 4 .

Total time 44.9425 s.

iteration number	time per iteration	new currents and powers	new terms of linearized sys.	sys. solution and new volts.
1	13.4295	0.1494	11.2954	1.9847
2-3	2.1488	0.1481	0.0112	1.9896
4 (ISW)	7.7244	0.1476	0.0112	7.5657
5 (N-R)	19.3418	0.1496	11.2930	7.8992
6	0.1493	0.1493	0.0	0.0

The same N.B. as in table 6.3 applies here.

The effectiveness of the exact solution of the linear system may be appreciated. It brings the voltage error almost to zero in a relatively small time. Newton-Raphson's method needs then only one iteration to finish off. The time saved with respect to the Newton-Raphson method is in this case 12.9 seconds (22.3%). The proposed method without switching to an exact solution, plus the Newton-Raphson iteration, takes 56.9 seconds to converge, i.e.: only saves 0.9 seconds in this case, with respect to the plain Newton-Raphson method, which takes 57.8 seconds.

As an example here are values of the number of runs per iteration and the iteration at which to solve the linear system in an exact way, and to switch to the Newton-Raphson method, so that an important saving of time is obtained. For these examples, no acceleration factors in the solution of the linearized system were used.

- for the 14 bus test system

$\begin{cases} 3 \text{ runs/ite.} \\ \text{ISW} = 3 \end{cases}$	save 16.00%	and	$\begin{cases} 2 \text{ runs/ite.} \\ \text{ISW} = 3 \end{cases}$	save 19.8%
---	-------------	-----	---	------------

- for the 30 bus test system

$\begin{cases} 2 \text{ runs/ite.} \\ \text{ISW} = 4 \end{cases}$	save 25.08%	and	$\begin{cases} 3 \text{ runs/ite.} \\ \text{ISW} = 3 \end{cases}$	save 26.0%
---	-------------	-----	---	------------

- for the 57 bus test system

$\begin{cases} 6 \text{ runs/ite.} \\ \text{ISW} = 5 \end{cases}$	save 18.85%	and	$\begin{cases} 12 \text{ runs/ite.} \\ \text{ISW} = 3 \end{cases}$	save 20.55%
---	-------------	-----	--	-------------

These are examples that have just been tested. Other values could give better results. In general, with the initial values used ($E_k^0 = 1.0$, $F_k^0 = -0.3$), the 30 bus test system tends to yield more convergent results.

VIII. CONCLUSIONS

8.1 INCONVENIENCES AND ADVANTAGES OF THE PROPOSED METHOD

A method has been described for solving the load-flow problems. It has been tested with the IEEE 14, 30, and 57 bus test systems. With all three systems, the proposed method gives the solution in less time than the Newton-Raphson method using triangulation with pivoting.

The proposed method is convergent for an initial guess of voltages if the particular guess yields convergence for the Newton-Raphson method.

When the proposed method is mentioned in this section, it means the proposed method as it is employed in section 7.4, that is to say, with an iterative approximate solution at the early stages, and with a switch to an exact solution and the Newton-Raphson method at the end of the process. There may exist other ways to implement the proposed method yielding more time improvement. Some alternative ways are described in section 8.3. In this thesis it is not intended to come up with the best implementation of the proposed method, but an attempt has been made to show that there are ways of implementating the proposed method that yield the solution in less time than the Newton-Raphson method.

Since the proposed method is a direct method, it has all the advantages and disadvantages of the direct methods over the bus by bus methods, (see section 1.2.1), so far as required core and

computer time taken by the solution are concerned.

The main advantage of the proposed method over the Newton-Raphson method is that the proposed method requires less time. The proposed method needs three extra vectors, whose dimensions are $2(NB - S)$, where NB stands for the number of buses and S is the number of slack buses. In the 57 bus test system these three vectors mean less than two per cent of the total core. Moreover, two of the three vectors are needed as workspace in the acceleration technique for the iterative solution of the linearized system. However, a different technique may not demand the use of the three vectors.

If several load-flow problems had to be solved for the same power system, each having a different set of scheduled powers, the proposed method could be used in a very efficient way, provided that the same set of initial voltages was used for each load-flow problem. The fact of not having to change the non-diagonal terms of the matrix of coefficients of the linearized system, which takes more than half of the time of the iteration if the change takes place, means that an important reduction of time can be achieved.

The switch to the exact solution in the proposed method must be done at the suitable iteration. It has not been proven that a criteria based on the value of the maximum power mismatch, or the maximum voltage change, or both, will be suitable for every power system. Therefore, when the proposed method is applied to a load-flow problem for the first time, a risk exists of switching before it is convenient, so ending up running more than one iteration with

the Newton-Raphson technique. This would mean in general, more time than the Newton-Raphson method alone.

If the maximum power mismatch or maximum voltage change criteria fails to give the right iteration for switching to the exact solution, a trial and error process could find it.

Another inconvenience lies in the fact that, when using the Gauss-Seidel technique for the solution of the linearized system, the problem is more sensitive to the initial guess than it is when the Newton-Raphson method is used.

8.2 REAL LOAD-FLOW PROGRAMS AND RESEARCH LOAD-FLOW PROGRAMS

The research load-flow programs handle a small number of buses in comparison with the real load-flow programs. Besides, in order to squeeze the real-load flow program into a non-prohibitive amount of memory, many logical instructions are used in the program so that only the non-zero elements of the matrix of coefficients of the linearized system, and bus impedance or admittance matrices are stored in the memory. This makes the solution of the load-flow problem longer, therefore a compromise must be reached between the amount of core taken, and the time for solution. The compromise chosen is different for different programs. Hence a fair comparison is difficult to draw between programs. Besides there are other special calculations such as voltage controlled buses, acceleration techniques, net exchange, etc., which make the comparison even more difficult. In the research load-flow programs, the tie line interchange, tap changing under load transformers, and com-

plicated logical programming aimed at saving memory, are usually disregarded.

In the proposed method, which has been tested with the IEEE 14, 30, and 57 bus test system, some logical instructions have been introduced so as to be able to deal with the 57 bus test system without special permit, i.e.: keeping the size of the program under 200K (OS IBM 360 system), which is the limit core capacity for an individual user of the computer of the University of Alberta. So, the program works only with the non-zero elements of the bus admittance matrix, which is very sparse. This brings the program employed here closer to the real-load flow programs. The logical modifications introduced changed the amount of time needed for the solution, however the increasing or decreasing trends of the percentages of time taken by each part of the iteration remained the same.

Although the results obtained using the proposed method are faster than those obtained with the Newton-Raphson method, it can not be concluded automatically that the proposed method is better because only the experience with a real load-flow problem can confirm the above mentioned conclusion. As the Newton-Raphson and the proposed method are similar, it is hoped that, when adapting the proposed method for a real load-flow problem, the same or similar accelerating techniques, and other special features, can be used without difficulty.

8.3 SUGGESTIONS FOR FURTHER RESEARCH ON THE PROPOSED METHOD

By taking advantage of not having to calculate the non-diagonal terms of the matrix of coefficients of the linearized system, many techniques aimed at improving the efficiency of the approximate solution of the linearized system at each iteration, could be implemented.

Improvement can be brought about by taking a variable number of runs per iteration and increasing the number of runs as the voltages get closer to the final values. The criteria for increasing the number of runs per iteration could be a prestablished one, or one based on the values of the maximum mismatch of power or the maximum change of voltage. This would certainly increase the efficiency of the calculation, and hence it would yield further time savings.

The exact solution of the linearized system by the triangulation with pivoting technique, which is a destructive method, when switching from an approximate solution technique to an exact one, could be replaced by a non destructive technique. As a non-destructive technique, the Gauss-Seidel iterative method could be used. The exact solution may take more time than triangulation with pivoting, because of the slow convergence of the Gauss-Seidel technique for the type of equation that the linearized system is represented by. On the other hand, the coefficients of the linearized system would not be destroyed, and two successive iterations with an exact solution would most likely lead to the final solution.

The acceleration formula for the iterative solution of the linearized system, can be made more efficient, mainly after the early iterations, when the results of successive runs do not oscillate. If the acceleration formula were more efficient, further savings of time would be obtained.

Either triangulation with pivoting or the Gauss-Seidel technique, could be used for the exact solution of the linearized system after the switch from the approximate solution. Then the coefficients of the linearized system could be recalculated, regarding the last calculated voltages as the new initial guesses. Once this is done, the linearized system is solved with the Gauss-Seidel technique using as many iterations as necessary to guarantee an accuracy within the tolerance.

The process described above is equivalent to the switch to an exact solution, and an iteration of the Newton-Raphson method. In the suggested way however, at the end of the problem, a new matrix of coefficients of the linearized system is available. This matrix has been calculated with an initial guess that is much closer to the solution than the one that was used before.

REFERENCES

- (1) M. A. LAUGHTON and M. W. HUMPHREY-DAVIES,
"Numerical techniques in solution of power-system load-flow problems". Proc. IEE (London), vol. 111, p. 1575, September 1964
- (2) A. M. SASSON and F. J. JAIMES,
"Digital methods applied to power flow studies". IEEE Trans. Power Apparatus and Systems, vol. 86, p.860, July 1967
- (3) G. W. STAGG and A. H. EL-ABIAD,
"Computer methods in power system analysis", (book). McGraw-Hill 1968
- (4) A. M. SASSON,
"Nonlinear programming solutions for load-flow, minimum-loss, and economic dispatch problems". IEEE Trans. Power Apparatus and Systems, vol. 88, p. 399, April 1969
- (5) J. B. WARD and H. W. HALE,
"Digital computer solution of power flow problems". Trans. AIEE Power Apparatus and Systems, vol. 75, p. 398, June 1956
- (6) W. F. TINNEY and C. E. HART,
"Power flow solution by Newton's method". IEEE Trans. Power Apparatus and Systems, vol. 86, p. 1499, November 1967
- (7) D. FEINGOLD and D. SPOHN,
"Les problèmes de répartition de puissance dans les réseaux maillés à courant alternatif. Analyse et méthodes numériques". Revue Générale de l'Electricité, vol. 76, p. 681, April 1967
- (8) J. E. VAN NESS,
"Iteration methods for load-flow studies". Trans. AIEE Power Apparatus and Systems, vol. 78, p. 583, August 1959
- (9) L. L. FRERIS and A. M. SASSON,
"Investigation of the load-flow problem", Proc. IEE (London), vol. 115, p. 1459, October 1968
- (10) P. HENRICI,
"Elements of numerical analysis", (book). John Wiley and Sons 1964
- (11) E. ISAACSON and H. B. KELLER,
"Analysis of numerical methods", (book). John Wiley and Sons 1966
- (12) S. D. CONTE,
"Elementary numerical analysis", (book). McGraw-Hill 1965
- (13) H.S. WALL
"A modification of Newton's method", American Mathematical Monthly, vol. 55, p. 90, 1948.

APPENDIX 1. AN ITERATIVE METHOD FOR THE SOLUTION OF A SET OF QUADRATIC NONLINEAR EQUATIONS

For the purpose of illustration, consider a very simple two dimensional example.

If we have a set of quadratic nonlinear equations (A1.1)

$$P = Aef$$

$$Q = Bef \tag{A1.1}$$

where e and f are the unknowns, and P, Q, A, B , are known quantities, a new set of equivalent equations, which will simplify the application of an iterative way of solution, can be derived.

Let e^* and f^* be the solution of the system of equations (A1.1), and let e^0 and f^0 be a pair of values which are approximations to the solutions. Then we can write

$$e^* = e^0 + \Delta e^0$$

$$f^* = f^0 + \Delta f^0 \tag{A1.2}$$

The solution of the system (A1.1) is e^*, f^* , which is equal to a starting value e^0, f^0 , plus the increments $\Delta e^0, \Delta f^0$, which brings the starting value to the true solution by (A1.2).

If $e^* = e^0 + \Delta e^0$, and $f^* = f^0 + \Delta f^0$, are substituted in (A1.1), this gives,

$$P = A(e^0 + \Delta e^0)(f^0 + \Delta f^0) = Ae^0f^0 + Ae^0\Delta f^0 + Af^0\Delta e^0 + A\Delta e^0\Delta f^0$$

$$Q = B(e^0 + \Delta e^0)(f^0 + \Delta f^0) = Be^0f^0 + Be^0\Delta f^0 + Bf^0\Delta e^0 + B\Delta e^0\Delta f^0$$

$$\tag{A1.3}$$

Calling $P^0 = Ae^0f^0$, $Q^0 = Be^0f^0$, and $\Delta P = P - P^0$, $\Delta Q = Q - Q^0$, and gathering terms in the right hand side in one of the possible ways, we get (A1.4)

$$\begin{aligned} P - P^0 &= \Delta P = A(f^0 + \Delta f^0)\Delta e^0 + Ae^0\Delta f^0 \\ Q - Q^0 &= \Delta Q = Bf^0\Delta e^0 + B(e^0 + \Delta e^0)\Delta f^0 \end{aligned} \quad (A1.4)$$

where the only unknowns are Δe^0 and Δf^0 .

The following iterative procedure can be used for finding Δe^0 and Δf^0 . At first we drop Δe^0 and Δf^0 before e^0 and f^0 in the right hand side of the equations (A1.4). It results in a linear system (A1.5)

$$\begin{aligned} \Delta P &= Af^0\Delta e + Ae^0\Delta f \\ \Delta Q &= Bf^0\Delta e + Be^0\Delta f \end{aligned} \quad (A1.5)$$

Once the system is solved, we obtain Δe^1 , Δf^1 , which are not the exact values of the solution Δe^0 , Δf^0 , because (A1.5) is only an approximation to the original system. If we now substitute Δe^1 , Δf^1 back in the diagonal coefficients of (A1.4) we get a new linear system whose coefficients are closer to those of (A1.4) than those of (A1.5), so an improved solution can be expected from the new linearized system. The solution of the new system, Δe^2 , Δf^2 , is substituted back again in the diagonal coefficients of (A1.4), and the procedure is repeated until the improvement in the values of Δe , Δf , is less than a certain error.

In general, at the "ith" iteration,

$$\Delta P = A(f^0 + \Delta f^i) \Delta e + A e^0 \Delta f$$

$$\Delta Q = B f^0 \Delta e + B(e^0 + \Delta e^i) \Delta f \quad (A1.6)$$

and the result would be Δe^{i+1} , Δf^{i+1} , which would be closer to Δe^0 , Δf^0 than Δe^i , Δf^i , because the coefficients of Δe and Δf in (A1.6) are closer to their true values, $A(f^0 + \Delta f^0)$, $B(e^0 + \Delta e^0)$, than those used in the former iteration.

It is obvious that if we had (A1.7) instead of (A1.1),

$$P = A_{1ee} + A_{2ef} + A_{3ff}$$

$$Q = B_{1ee} + B_{2ef} + B_{3ff} \quad (A1.7)$$

or any number of equations, with the same number of unknowns, we would always be able to apply the same method of solution, provided that the equations were quadratic. See section 3.1.1 .

APPENDIX 2. THE CALCULATION OF THE Y_{bus} MATRIX

For each test power system, we have as data, the line impedance, and the line charging admittance. We have to compute the bus admittance matrix, Y_{bus} .

There are some transformers with off-nominal turn ratios, and synchronous condensers connected to some of the buses. The synchronous condensers can be replaced by a susceptance.

The formulas are,

$$y_{ij} = - \frac{1}{z_{ij}} \quad i \neq j \quad (A2.1)$$

$$y_{ii} = \sum_{j \neq i} \frac{1}{z_{ij}} + \sum_{j \neq i} \frac{y'_{ij}}{2} + c_i \quad (A2.2)$$

where z_{ij} are the line impedance, y_{ij} , the bus admittances, $y'_{ij}/2$ is the one half of total line charging admittances, and c_i the synchronous condenser susceptance, if it exists, at the bus "i".

If between bus "p" and bus "q", there is a transformer with an off-nominal turn ratio "a", then,

$$y_{pq} = - \frac{1}{axz_{pq}} \quad p \neq q \quad (A2.3)$$

$$y_{qp} = - \frac{1}{axz_{qp}} \quad q \neq p \quad (A2.4)$$

$$y_{pp} = \sum_{\substack{r \neq p \\ r \neq q}} \frac{1}{z_{pr}} + \frac{1}{a^2 xz_{pq}} + \sum_{q \neq p} \frac{y'_{pq}}{2} + c_p \quad (A2.5)$$

$$y_{qq} = \sum_{\substack{p \\ p \neq q}} \frac{1}{z_{qp}} + \sum \frac{y'_{pq}}{2} + c_q \quad (\text{A2.6})$$

The subroutine used for the calculations yields only the non-zero elements of Y_{bus} that are above the diagonal, with a double array of numbers that give the ordered numbers of pairs of buses that correspond to the nonzero elements of Y_{bus} .

APPENDIX 3. THE SUBROUTINE FOR THE IMPLEMENTATION OF THE PROPOSED METHOD AND THE NEWTON-RAPHSON METHOD

Before the flow-charts of the Newton-Raphson method and the proposed method, we will give a list of symbols:

e_k^0, f_k^0	initial or estimated voltage at node "k"
P_k^0, Q_k^0	powers calculated with the voltages e_i^0, f_i^0 according to the formula (1.5)
$\Delta e_k, \Delta f_k$	increment of voltage between two successive iterations of the Newton-Raphson method
$\Delta e_k, \Delta f_k$	difference between the voltage at the end of an iteration of the proposed method and the initial voltage e_k^0, f_k^0
$\Delta e_k^0, \Delta f_k^0$	difference between the true value of the voltages and the estimated ones
$\Delta P_k^0, \Delta Q_k^0$	difference between the scheduled power and that calculated with the initial values of the voltages
$\Delta P_k, \Delta Q_k$	difference between the scheduled power and that calculated with the iterated voltages
CIR_k, CII_k	current at the bus "k", when the voltages are e_k, f_k
CIR_k^0, CII_k^0	current at the bus "k", when the voltages are e_i^0, f_i^0
DCR_k^0, DCI_k^0	difference between the true value of the current at bus "k" and its initial estimate
CIR_k^0, CII_k^0	
j_{ik}	element of the jacobian in the Newton-Raphson method

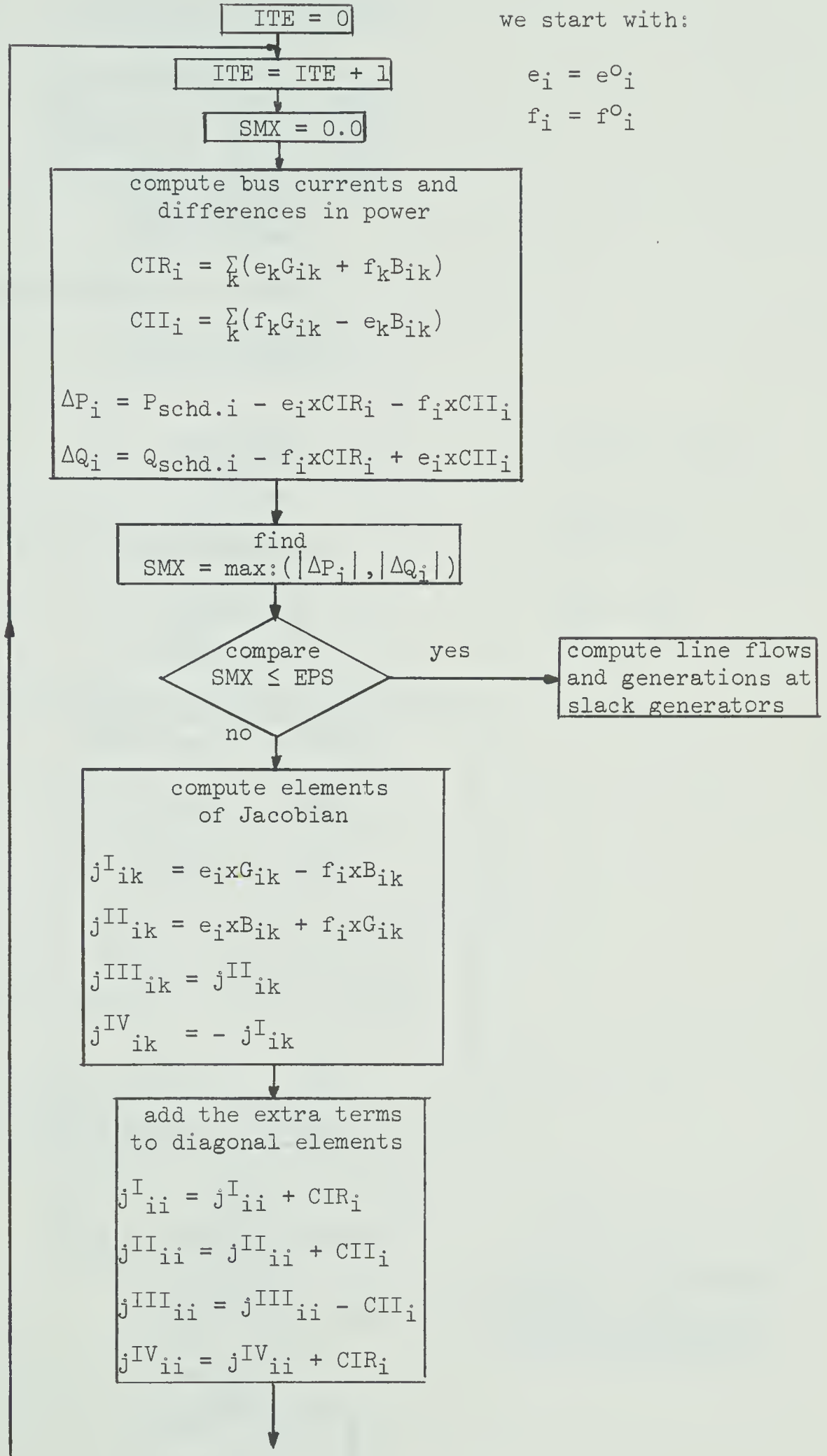
l_{ik}	element of the linearized system in the proposed method
SMX	maximum mismatch in power with respect to the scheduled powers
TMX	maximum change in voltage between the actual and the former iteration
EPS	tolerance or precision
ITE	number of the iteration
NR	constant which is 1 if the operations are made as in the Newton-Raphson method, and 0 if as in the proposed method

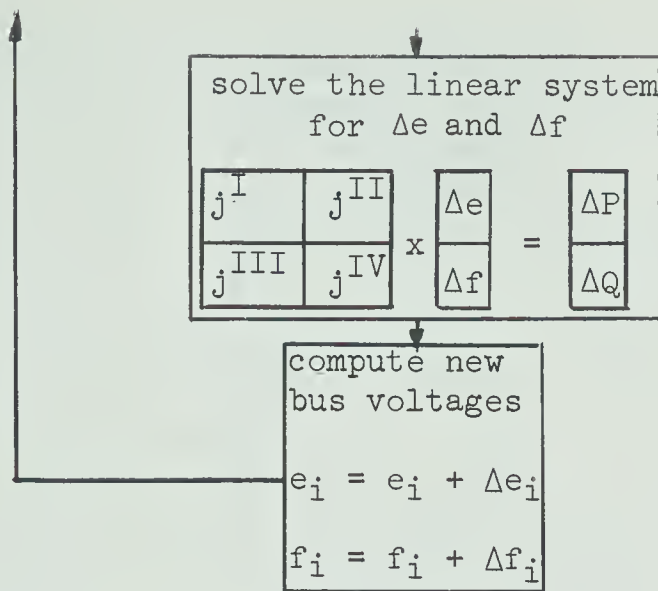
"true value" of a variable stands for the value that this variable has in the power system when it is operating with the scheduled powers.

Here follow some comments on the flow-charts and the program used.

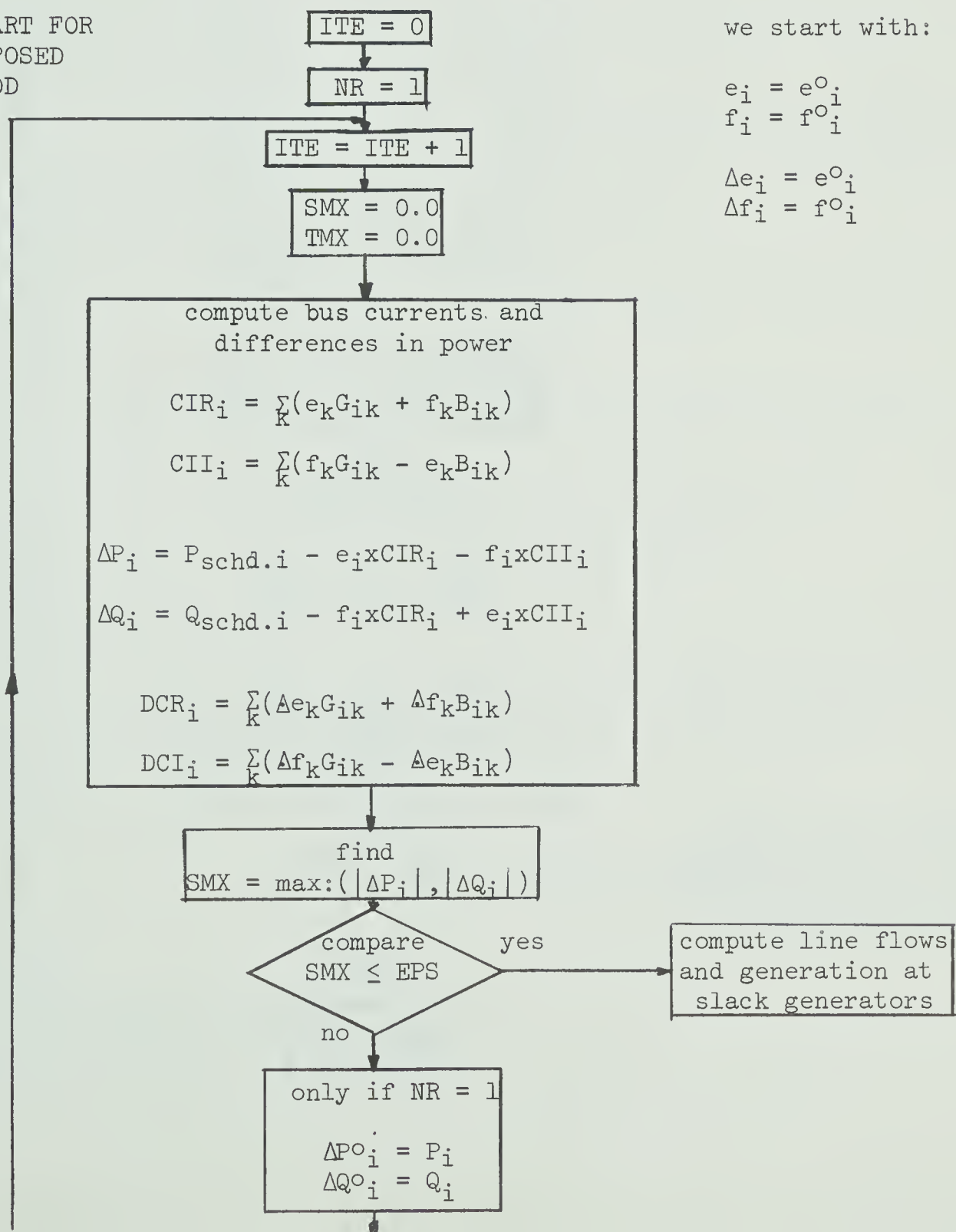
- A separate flow chart is given for the Newton-Raphson method and for the proposed method so that the differences can be clearly seen. However keeping NR equal to 1, the flow chart for the proposed method also works for the Newton-Raphson method.
- The voltage controlled buses are handled in the proposed method, in the same way as in the Newton-Raphson method. Details can be found in section 8.6 of (3).
- With the subroutine EDTM, which is shown below, we can effect either the Newton-Raphson method or the proposed method. Any bus can be the slack bus, and we can use any number of slack buses. (See appendix 6).
- To implement the proposed method all we have to do is to keep NR zero after the first iteration.

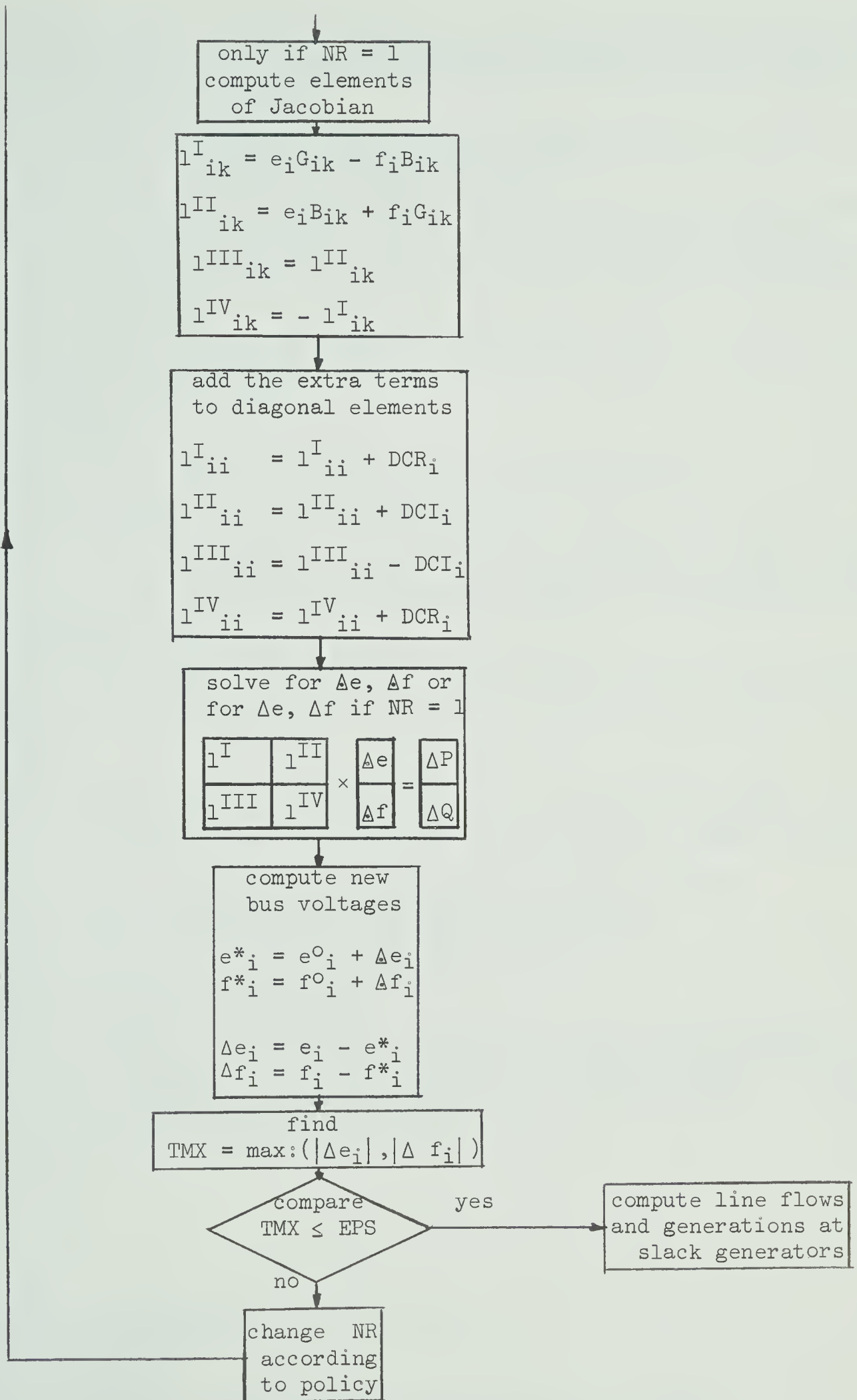
FLOW CHART OF
THE METHOD OF
NEWTON-RAPHSON





FLOW CHART FOR
THE PROPOSED
METHOD






```

1      SUBROUTINE EDTM (NJ,NB,EPS,FLUX,LL,IN,JN,E,F,KNS,NZ,IOR,QX,QN,J,G,
2      1B,PES,QES,DP,DQ,DCR,DCI,DFS,DFT,ERG,ATR,LEL,GRF,LRF,NS,ESR,FSR,
3      2EAC,FAC,IORL,JORL,JOR,INF,INFB,KPT,DPT,EF,FF,DEF,DFF,NGB,IGB,JGB,
4      3NTI,ICV,DY1,DY2,FAT,FTA)
5      REAL*8 J,G,B,PES,QES,DP,DQ,E,F,DCR,DCI,DFS,DFT,QX,QN,ERG,CP,CN,
6      1EPS,ESR,FSR,EAC,FAC,PP,PS,AUX,ZER,TMX,SMX
7      COMPLEX*16 EUX,UEX,ZEC,FLUX,GRF,ATR,PRD
8      DIMENSION J(NJ,NJ),G(NGB),B(NGB),PES(NB),QES(NB),DP(NB),DQ(NB),
9      1E(NB),F(NB),DCR(NB),DCI(NB),DFS(NJ),DFT(NJ),QX(NZ),QN(NZ),
10     2ERG(NZ),GRF(LRF),NS(LRF),FLUX(LL),ATR(LEL),JOR(NJ),IOR(NZ),
11     3IORL(LEL),JORL(LEL),IN(LL),JN(LL),ESR(NB),FSR(NB),EAC(NB),FAC(NB),
12     4IGB(NGB),JGB(NGB)
13     REAL*8 DPT,EF,FF,DEF,DFF,DY1,DY2,FAT,FTA
14     DIMENSION DPT(NJ),EF(NB),FF(NB),DEF(NB),DFF(NB),DY1(NJ),DY2(NJ)
15     DIMENSION TEMPS(100,3)
16 350 FORMAT (6E13.6)
17 365 FORMAT (////13X,4HITE ,12)
18 366 FORMAT (12X,12,1H-,12,2X,F10.5,3H +(,F10.5,2H)J)
19 367 FORMAT (////11X,'FINAL RESULT'//13X,'BUS',12X,'P-JQ',13X,'ITE=',
20     112,2X,'TIME USED ',F8.4,' SECONDS'/)
21 368 FORMAT (/13X,40HIMPOSSIBLE TO HOLD THE VOLTAGE IN NODE 12,3X,
22     13HQP=,F10.5/13X,46HFROM NOW ON, THIS BUS WILL HAVE CONSTANT POWER)
23 369 FORMAT (///23X,25HVOLTAGE CONTROLLED BUSES //12X,3HBUS,5X,4HECNT,
24     19X,4HQMAX,9X,4HQMIN/)
25 370 FORMAT (12X,12,2X,F10.5,3X,F10.5,3X,F10.5)
26 371 FORMAT (////////13X,'*****'//13X,'LOAD FLOW PROBLEM'//13
27     1X,17H*****')
28 372 FORMAT (///13X,25HVOLTAGE CONTROLLED BUSES //12X,3HBUS,5X,4HECNT,
29     111X,1HQ)
30 373 FORMAT (12X,12,2X,F10.5,3X,F10.5)
31 374 FORMAT (/13X,42H*** NO RESTRICTION ON REACTIVE POWER ***)
32 375 FORMAT (/13X,59H*** RESTRICTIONS ON REACTIVE POWER TAKEN INTO AC
33     1COUNT ***)
34 376 FORMAT (/11X,5HSLACK,1X,12,F10.5,3H +(,F10.5,2H)J)
35 377 FORMAT (/11X,6HLOSSES2X,F10.5,3H +(F10.5,2H)J)
36 378 FORMAT (/13X,'TIME USED ',F8.4,' SECONDS')
37 379 FORMAT (/13X,'BUS',12X,'G-JB',/)
38 380 FORMAT (/13X,'SMX',F10.5)
39 381 FORMAT (/13X,'TMX',F10.5)
40 382 FORMAT (///9X,'ITERATION',1X,'TIME PER',3X,'NEW CURRENTS',1X,'NEW
41     1TERMS',3X,'LINEAR SYS.',2X,'TOTAL TIME'/10X,'NUMBER',3X,'ITERATION
42     2',3X,'AND POWERS',1X,'IN JACOBIAN',1X,'AND NEW VOLS. SECONDS')
43 383 FORMAT (/12X,12,1X,5(F12.4),1X)
44     ZER=0.0
45     ZEC=(0.0,0.0)
46     KZ=0
47     KU=1
48     KD=2
49     NTT=0
50     KFI=0
51     MT=240*(1000000/26)
52     WRITE (6,371)
53     WRITE (6,379)
54     WRITE (6,366) ((IGB(I),JGB(I),G(I),B(I)),I=1,NGB)

```



```

55      IF (INF.EQ.0) GO TO 501
56      CALL EDVR (KU,NB,NB,NS,LRF,PES,QES,4H BUS,4H PES,4H QES)
57      CALL EDVR (KU,NB,NB,NS,LRF,E,F,4H BUS,4H EES,4H FES)
58      IF (KNS.EQ.0) GO TO 501
59      WRITE (6,369)
60      WRITE (6,370) ((IOR(K),ERG(K),QX(K),QN(K)),K=1,NZ)
61      WRITE (6,374)
62  501  ITE=0
63      KDD=0
64      IF (KPT.EQ.1) KDD=1
65      CALL CS019A (MT)
66  499  NR=1
67      DO 500 I=1,NB
68          EAC(I)=E(I)
69          FAC(I)=F(I)
70          ESR(I)=E(I)
71  500  FSR(I)=F(I)
72  502  ITE=ITE+1
73      IF (INFB.NE.0) WRITE (6,365) ITE
74      IF (ITE.GE.ICV) KDD=0
75      SMX=0.0
76      LI=1
77      DO 506 I=1,NB
78          IF (I.NE.NS(LI)) GO TO 503
79          IF (LI.LT.LRF) LI=LI+1
80      GO TO 506
81  503  PP=0.0
82      PS=0.0
83      CP=0.0
84      CN=0.0
85      KSY=0
86      DO 504 K=1,NGB
87          IF (IGB(K).GT.I.AND.KSY.EQ.1) GO TO 536
88          IF (I.NE.JGB(K)) GO TO 537
89          PP=PP+E(IGB(K))*G(K)+F(IGB(K))*B(K)
90          PS=PS+F(IGB(K))*G(K)-E(IGB(K))*B(K)
91          CP=CP+EAC(IGB(K))*G(K)+FAC(IGB(K))*B(K)
92          CN=CN+FAC(IGB(K))*G(K)-EAC(IGB(K))*B(K)
93      GO TO 504
94  537  IF (I.NE.JGB(K)) GO TO 504
95          PP=PP+E(JGB(K))*G(K)+F(JGB(K))*B(K)
96          PS=PS+F(JGB(K))*G(K)-E(JGB(K))*B(K)
97          CP=CP+EAC(JGB(K))*G(K)+FAC(JGB(K))*B(K)
98          CN=CN+FAC(JGB(K))*G(K)-EAC(JGB(K))*B(K)
99      KSY=1
100  504  CONTINUE
101  536  DCR(I)=CP
102      DCI(I)=CN
103      AUX=PES(I)-E(I)*PP-F(I)*PS
104      IF (NR.EQ.1) DP(I)=AUX
105      IF (DABS(AUX).GT.SMX) SMX=DABS(AUX)
106      AUX=QES(I)-F(I)*PP+E(I)*PS
107      IF (KNS.EQ.0) GO TO 533
108      DO 505 K=1,NZ

```



```
109      IF (I.EQ.IOR(K)) AUX=ERG(K)**2-E(I)**2-F(I)**2
110      IF (I.EQ.IOR(K)) DQ(I)=AUX
111 505 CONTINUE
112 533 IF (NR.EQ.1) DQ(I)=AUX
113      IF (DABS(AUX).GT.SMX) SMX=DABS(AUX)
114 506 CONTINUE
115      IF (INFB.NE.0) WRITE (6,380) SMX
116      IF (SMX.GT.EPS) GO TO 534
117      CALL CS019B (MTR)
118      TEMPS(ITE,3)=FLOAT(MT-MTR)*2.6E-05
119      NTT=1
120      GO TO 518
121 534 IF (INFB.NE.0.AND.NR.EQ.1) CALL EDVR (KZ,NB,NB,NS,LRF,DP,DQ,' BUS '
122      1,'DELP','DELQ')
123      CALL CS019B (MTR)
124      TEMPS(ITE,1)=FLOAT(MT-MTR)*2.6E-05
125      I=0
126      LI=1
127      DO 515 II=1,NB
128      IF (II.NE.NS(LI)) GO TO 507
129      IF (LI.LT.LRF) LI=LI+1
130      GO TO 515
131 507 I=I+1
132      INL=I+NB-LRF
133      IF (NR.NE.1) GO TO 512
134      DFS(I)=DP(II)
135      DFS(INL)=DQ(II)
136      K=0
137      LK=1
138      DO 511 KK=1,NB
139      IF (KK.NE.NS(LK)) GO TO 508
140      IF (LK.LT.LRF) LK=LK+1
141      GO TO 511
142 508 K=K+1
143      KNL=K+NB-LRF
144      J(I,K)=0.0
145      J(I,KNL)=0.0
146      DO 550 N=1,NGB
147      IF ((IGB(N).NE.II.OR.JGB(N).NE.KK).AND.(IGB(N).NE.KK.OR.JGB(N).NE
148      111)) GO TO 550
149      J(I,K)=E(II)*G(N)-F(II)*B(N)
150      J(I,KNL)=E(II)*B(N)+F(II)*G(N)
151      GO TO 549
152 550 CONTINUE
153 549 IF (KNS.EQ.0) GO TO 510
154      MS=0
155      DO 509 M=1,NZ
156      IF (II.NE.IOR(M)) GO TO 509
157      J(INL,K)=0.0
158      J(INL,KNL)=0.0
159      MS=1
160 509 CONTINUE
161      IF (MS.EQ.1) GO TO 511
162 510 J(INL,K)=J(I,KNL)
```



```
163      J(INL,KNL)=-J(I,K)
164 511 CONTINUE
165 512 J(I,I)=J(I,I)+DCR(II)
166      J(I,INL)=J(I,INL)+DCI(II)
167      IF (KNS.EQ.0) GO TO 514
168      MS=0
169      DO 513 M=1,NZ
170      IF (II.NE.IOR(M)) GO TO 513
171      J(INL,I)=ESR(II)+E(II)
172      J(INL,INL)=FSR(II)+F(II)
173      MS=1
174 513 CONTINUE
175      IF (MS.EQ.1) GO TO 515
176 514 J(INL,I)=J(INL,I)-DCI(II)
177      J(INL,INL)=J(INL,INL)+DCR(II)
178 515 CONTINUE
179      CALL CS019B (MTR)
180      TEMPS(ITE,2)=FLOAT(MT-MTR)*2.6E-05
181      IF (KDD.EQ.0) CALL RSLT (J,DFS,DFT,JOR,NJ)
182      IF (KDD.NE.0) CALL GSDD (J,DFS,DFT,DPT,NJ,NS,LRF,NB,NTI,DY1,DY2,
183 1FAT,FTA,ITE)
184      TMX=0.0
185      I=0
186      LI=1
187      DO 517 II=1,NB
188      IF (II.NE.NS(LI)) GO TO 516
189      FAC(II)=0.0
190      EAC(II)=0.0
191      IF (LI.LT.LRF) LI=LI+1
192      GO TO 517
193 516 I=I+1
194      INL=I+NB-LRF
195      EAC(II)=ESR(II)+DFT(I)-E(II)
196      FAC(II)=FSR(II)+DFT(INL)-F(II)
197      IF (DABS(EAC(II)).GT.TMX) TMX=DABS(EAC(II))
198      IF (DABS(FAC(II)).GT.TMX) TMX=DABS(FAC(II))
199      E(II)=ESR(II)+DFT(I)
200      F(II)=FSR(II)+DFT(INL)
201      DEF(II)=EF(II)-E(II)
202      DFF(II)=FF(II)-F(II)
203 517 CONTINUE
204      IF (INFB.NE.0) WRITE (6,381) TMX
205      IF (INFB.NE.0) CALL EDVR (KZ,NB,NB,NS,LRF,DEF,DFF,' BUS',' DEF','
206 1DFF')
207      CALL CS019B (MTR)
208      TEMPS(ITE,3)=FLOAT(MT-MTR)*2.6E-05
209      IF (TMX.LE.EPS) GO TO 518
210      IF (INFB.NE.0) WRITE (6,378) TEMPS(ITE,3)
211      NR=0
212      IF (KDD.EQ.0) GO TO 499
213      IF (KDD.NE.0) GO TO 502
214 518 WRITE (6,367) ITE,TEMPS(ITE,3)
215      DO 521 I=1,LL
216      UEX=ZEC
```



```

217      DO 519 K=1,NGB
218      IF (IN(I).EQ.IGB(K).AND.JN(I).EQ.JGB(K).OR.IN(I).EQ.JGB(K).AND.
219      1JN(I).EQ.IGB(K)) UEX=-DCMPLX(G(K),-B(K))
220 519 CONTINUE
221      EUX=ZEC
222      DO 520 K=1,LEL
223      IF (IN(I).EQ.IORL(K).AND.JN(I).EQ.JORL(K).OR.IN(I).EQ.JORL(K).AND.
224      1JN(I).EQ.IORL(K)) EUX=ATR(K)
225 520 CONTINUE
226      FLUX(I)=(E(IN(I))*2+F(IN(I))*2)*(UEX+EUX)+UEX*DCMPLX((-E(IN(I))*
227      1E(JN(I))-F(IN(I))*F(JN(I))), (E(JN(I))*F(IN(I))-E(IN(I))*F(JN(I))))
228 521 WRITE (6,366) IN(I),JN(I),FLUX(I)
229      PRD=ZEC
230      DO 524 L=1,LRF
231      NSL=NS(L)
232      EUX=ZEC
233      DO 522 I=NSL,LL
234      IF (IN(I).LT.NSL) GO TO 522
235      IF (IN(I).EQ.NSL) EUX=EUX+FLUX(I)
236      IF (IN(I).GT.NSL) GO TO 523
237 522 CONTINUE
238 523 WRITE (6,376) NSL,EUX
239      PRD=PRD+EUX
240 524 GRF(L)=EUX
241      DO 525 I=1,NB
242      PRD=PRD+DCMPLX(PES(I),-QES(I))
243 525 CONTINUE
244      WRITE (6,377) PRD
245      CALL EDVR (KU,NB,NB,NS,LRF,E,F,4H BUS,4HEFIN,4HFFID)
246 551 WRITE (6,382)
247      DO 552 I=1,ITE
248      IF (I.EQ.1) TM0=TEMPS(I,3)
249      IF (I.GE.2) TM0=TEMPS(I,3)-TEMPS(I-1,3)
250      IF (NTT.EQ.1.AND.I.EQ.ITE) GO TO 553
251      TM3=TEMPS(I,3)-TEMPS(I,2)
252      TM2=TEMPS(I,2)-TEMPS(I,1)
253      IF (I.GE.2) TM1=TEMPS(I,1)-TEMPS(I-1,3)
254      IF (I.EQ.1) TM1=TEMPS(I,1)
255      IF(I.LT.ITE.OR.NTT.EQ.0)WRITE(6,383) I, TM0, TM1, TM2, TM3, TEMPS(I,3)
256 553 IF(NTT.EQ.1.AND.I.EQ.ITE)WRITE(6,383) I, TM0, TM0, ZER, ZER, TEMPS(I,3)
257 552 CONTINUE
258      IF (KNS.EQ.0) RETURN
259      WRITE (6,372)
260      DO 527 K=1,NZ
261      IF (IOR(K).EQ.0) GO TO 527
262      PP=0.0
263      PS=0.0
264      KSY=0
265      DO 526 I=1,NGB
266      IF (IGB(I).GT.IOR(K).AND.KSY.EQ.1) GO TO 532
267      IF (IOR(K).NE.JGB(I)) GO TO 535
268      PP=PP+E(IGB(I))*G(I)+F(IGB(I))*B(I)
269      PS=PS+F(IGB(I))*G(I)-E(IGB(I))*B(I)
270      GO TO 526

```



```
271 535 IF (IOR(K).NE.IGB(I)) GO TO 526
272 PP=PP+E(JGB(I))*G(I)+F(JGB(I))*B(I)
273 PS=PS+F(JGB(I))*G(I)-E(JGB(I))*B(I)
274 KSY=1
275 526 CONTINUE
276 532 DQ(IOR(K))=F(IOR(K))*PP-E(IOR(K))*PS-QES(IOR(K))
277 AUX=DSQRT(E(IOR(K))**2+F(IOR(K))**2)
278 WRITE (6,373) IOR(K),AUX,DQ(IOR(K))
279 527 CONTINUE
280 KFI=0
281 DO 531 K=1,NZ
282 IF (IOR(K).EQ.0) GO TO 531
283 IF (DQ(IOR(K)).GE.QX(K)) GO TO 528
284 IF (DQ(IOR(K)).LE.QN(K)) GO TO 529
285 GO TO 531
286 528 QES(IOR(K))=QES(IOR(K))+QX(K)
287 GO TO 530
288 529 QES(IOR(K))=QES(IOR(K))+QN(K)
289 530 WRITE (6,368) IOR(K),QES(IOR(K))
290 IOR(K)=0
291 KFI=1
292 531 CONTINUE
293 IF (KFI.EQ.0) RETURN
294 WRITE (6,375)
295 IF (INFB.NE.0) CALL EDVR (KU,NB,NB,NS,LRF,EAC,FAC,4H BUS,4H EES,4H
296 1 FES)
297 GO TO 501
298 END
```


APPENDIX 4. OTHER SUBROUTINES USED

A number of subroutines are used to perform special operations during the execution or the printing operations of the load-flow problem. They are the following:

EDVR which prints two vectors in two columns side by side, along with the numbers of the components of the vectors.

RSLT which solves a linear system by means of triangulation with pivoting technique

GSDD which solves a linear system by means of the Gauss-Seidel technique with the modifications explained in section 5.1.2.1 .

Any number of runs per iteration can be used

FOAC which implements the acceleration technique explained in section 7.1.1 in the solution of the linearized system by means of the subroutine GSDD

All the programs have been written in Fortran IV, and all the variables and workspaces of all the subroutines have adjustable dimensions, so that the dimensions of the arrays and matrices, have to be written only once in the main program.

APPENDIX 5. THE SOLUTION OF THE LOAD-FLOW PROBLEMS

The final voltages, and scheduled powers of the 14, 30, and 57 bus test systems are shown next. The values are those obtained after the trial and error process by which it is determined whether the voltage magnitude at the voltage controlled buses can be held at the specified values by means of the available reactive power at these buses. In section 4.1.1 it is explained that to measure the properties of the method, only the first stage of the trial and error technique is necessary.

Only in the case of the 57 bus test system are we able to reach the final solution at the end of the first trial, as far as availability of reactive power at the voltage controlled buses, is concerned. In the 14 and 30 bus test systems we need more than one trial to arrive at the solutions shown in this appendix.

We will call the real power, P , the reactive power, Q , the real voltage, E , and the reactive voltage, F . The voltage controlled buses will be signaled, and there will be an indication whether the scheduled voltage magnitude can be held or it is impossible to do so with the available reactive power.

RESULTS OF THE 14 BUS TEST SYSTEM

	bus	P	Q	E	F	controlled vol. magn.
slack	1	2.33499	-0.43432	1.06000	0.00000	
vl. cont.	2	0.18300	-0.18978	1.01658	-0.08340	1.02000
vl. cont.	3	-0.94200	-0.19000	0.92885	-0.20713	impossible
	4	-0.47800	0.03900	0.96111	-0.17252	
	5	-0.07600	-0.01600	0.97284	-0.14721	
vl. cont.	6	-0.11200	-0.01500	0.97903	-0.25454	impossible
	7	0.00000	0.00000	0.97190	-0.23418	
vl. cont.	8	0.00000	0.00000	0.98208	-0.23663	impossible
	9	-0.29500	-0.16600	0.95986	-0.26304	
	10	-0.09000	-0.05800	0.95418	-0.26484	
	11	-0.03500	-0.01800	0.96227	-0.26124	

	bus	P	Q	E	F	controlled vol. magn.
	12	-0.06100	-0.01600	0.95939	-0.26659	
	13	-0.13500	-0.05800	0.95399	-0.26663	
	14	-0.14900	-0.05000	0.93356	-0.27793	
losses		0.14499	-0.18068			

RESULTS OF THE 30 BUS TEST SYSTEM

	bus	P	Q	E	F	controlled vol. magn.
slack	1	2.60989	-0.16100	1.06000	0.00000	
vl. cont.	2	0.18300	0.37300	1.03801	-0.09988	impossible
	3	-0.02400	-0.01200	1.01057	-0.14187	
	4	-0.07600	-0.01600	0.99731	-0.16952	
vl. cont.	5	-0.94200	0.18104	0.97829	-0.25111	1.01000
	6	0.00000	0.00000	0.99026	-0.19923	
	7	0.22800	-0.10900	0.97063	-0.22790	
vl. cont.	8	-0.30000	0.07576	0.98746	-0.21218	1.01000
	9	0.00000	0.00000	1.01742	-0.26166	
	10	-0.05800	-0.02000	1.00388	-0.28817	
vl. cont.	11	0.00000	0.16372	1.04790	-0.26950	1.08200
	12	-0.11200	-0.07500	1.01993	-0.27890	
vl. cont.	13	0.00000	0.10422	1.03307	-0.28249	1.07100
	14	-0.06200	-0.01600	1.00140	-0.29069	
	15	-0.08200	-0.02500	0.99665	-0.29115	
	16	-0.03500	-0.01800	1.00486	-0.28571	
	17	-0.09000	-0.05800	0.99801	-0.28957	
	18	-0.03200	-0.00900	0.98537	-0.29984	
	19	-0.09500	-0.03400	0.98264	-0.30256	
	20	-0.02200	-0.00700	0.98787	-0.30065	
	21	-0.17500	-0.11200	0.98979	-0.29247	
	22	0.00000	0.00000	0.99040	-0.29239	
	23	-0.03200	-0.01600	0.98429	-0.29462	
	24	-0.08700	-0.06700	0.97762	-0.29562	
	25	0.00000	0.00000	0.97581	-0.28703	
	26	-0.03500	-0.02300	0.95676	-0.28906	
	27	0.00000	0.00000	0.98414	-0.27965	
	28	0.00000	0.00000	0.98469	-0.20931	
	29	-0.02400	-0.00900	0.95895	-0.29488	
	30	-0.10600	-0.01900	0.94337	-0.30609	
losses		0.17589	-0.92300			

RESULTS OF THE 57 BUS TEST SYSTEM

	bus	P	Q	E	F	controlled vol. magn.
slack	1	4.23694	1.11801	1.04000	0.00000	
vl. cont.	2	-0.03000	-0.88754	1.00978	-0.02094	1.01000
vl. cont.	3	-0.01000	-0.18175	0.97963	-0.10276	0.98500
	4	0.00000	-0.21000	0.97275	-0.12526	

	bus	P	Q	E	F	controlled vol. magn.
	5	-0.13000	-0.04000	0.96566	-0.14512	
vl. cont.	6	-0.75000	0.05432	0.96879	-0.14781	0.98000
	7	0.00000	0.00000	0.97556	-0.13020	
vl. cont.	8	3.00000	0.40091	1.00193	-0.07847	1.00500
vl. cont.	9	-1.21000	-0.77919	0.96632	-0.16318	0.98000
	10	-0.05000	-0.02000	0.96665	-0.19579	
	11	0.00000	0.00000	0.95862	-0.17239	
vl. cont.	12	-0.67000	1.04518	0.99810	-0.18448	1.01500
	13	-0.18000	-0.02300	0.96463	-0.16671	
	14	-0.10500	-0.05300	0.95735	-0.15767	
	15	-0.22000	-0.05000	0.98030	-0.12369	
	16	-0.43000	-0.03000	1.00128	-0.15607	
	17	-0.42000	-0.08000	1.01295	-0.09568	
	18	-0.27200	-0.09800	0.97978	-0.20344	
	19	-0.03300	-0.00600	0.94449	-0.22203	
	20	-0.02300	-0.01000	0.93748	-0.22416	
	21	0.00000	0.00000	0.98310	-0.22575	
	22	0.00000	0.00000	0.98454	-0.22510	
	23	-0.06300	-0.02100	0.98290	-0.22590	
	24	0.00000	0.00000	0.97258	-0.22590	
	25	-0.06300	-0.03200	0.93365	-0.30650	
	26	0.00000	0.00000	0.93441	-0.21544	
	27	-0.09300	-0.00500	0.96184	-0.19594	
	28	-0.04600	-0.02300	0.98008	-0.18133	
	29	-0.17000	-0.02600	0.99558	-0.17147	
	30	-0.03600	-0.01800	0.91189	-0.30902	
	31	-0.05800	-0.02900	0.88304	-0.31070	
	32	-0.01600	-0.00800	0.90088	-0.30166	
	33	-0.03800	-0.01900	0.89849	-0.30156	
	34	0.00000	0.00000	0.93027	-0.23456	
	35	-0.06000	-0.03000	0.93806	-0.23231	
	36	0.00000	0.00000	0.94850	-0.23014	
	37	0.00000	0.00000	0.95806	-0.22912	
	38	-0.14000	-0.07000	0.98809	-0.22338	
	39	0.00000	0.00000	0.95587	-0.22939	
	40	0.00000	0.00000	0.94547	-0.22981	
	41	-0.06300	-0.03000	0.96637	-0.24233	
	42	-0.07100	-0.04400	0.93131	-0.25887	
	43	-0.02000	-0.01000	0.98984	-0.19879	
	44	-0.12000	-0.01800	0.99526	-0.20901	
	45	0.00000	0.00000	1.02257	-0.16693	
	46	0.00000	0.00000	1.04004	-0.20438	
	47	-0.29700	-0.11600	1.00888	-0.22392	
	48	0.00000	0.00000	1.00275	-0.22440	
	49	-0.18000	-0.08500	1.01016	-0.23209	
	50	-0.21000	-0.10500	0.99559	-0.23745	
	51	-0.18000	-0.05300	1.02726	-0.22837	
	52	-0.04900	-0.02200	0.96071	-0.19542	
	53	-0.20000	-0.10000	0.94884	-0.20607	
	54	-0.04100	-0.01400	0.97559	-0.20221	
	55	-0.06800	-0.03400	1.01253	-0.19318	
	56	-0.07600	-0.02200	0.93065	-0.26804	
	57	-0.06700	-0.02000	0.92480	-0.27545	
losses		0.27894	-2.07599			

APPENDIX 6. OPERATION WITH MORE THAN ONE SLACK BUS

One of the features of the subroutine EDTM is the possibility of using more than one slack bus, either for the Newton-Raphson method, or for the proposed method. A slack bus, as defined in section 1.2, is one where the two components of the voltage are kept constant. Once we know the load-flow solution, we can compute the generated or consumed power at this bus by means of the equation (1.5).

The calculated powers at the slack buses will balance the generations and loads at the other buses, and the losses in the power system.

If we solve the load-flow problem with only one slack bus, and then we solve the same problem with more than one slack bus, using as constant voltages at the slack buses the voltages obtained as solution of the load-flow with one slack bus, we are bound to obtain the same generations or loads at the slack buses and the same losses in the power system. Experimentation proves that to be true; only slight differences are found due to numerical errors.

Each set of different constant voltages for the slack buses brings about different power generations or consumptions at the slack buses and different losses in the power system.

The more slack buses are used, the more numerically stable is the problem. Some trial and error techniques could be tried on numerically instable load-flow by means of the use of more than one slack bus.

B29976